NetScaler 9.3

Contents

NetScaler 9.3

The Citrix NetScaler product line optimizes delivery of applications over the Internet and private networks, combining application-level security, optimization, and traffic management into a single, integrated appliance. You install a NetScaler appliance in your server room and route all connections to your managed servers through it. The NetScaler features that you enable and the policies you set are then applied to incoming and outgoing traffic.

What's New in NetScaler 9.3

The 9.3 release provides new wizards, including a wizard for configuring Citrix Application Firewall™, includes numerous enhancements of a wide variety of features, and delivers some performance enhancements. For a summary of the updates, see the Citrix NetScaler 9.3 Release Notes.

NetScaler Features

Important: We are in the process of transitioning product documentation to Citrix eDocs. This page includes links to product documentation that is located in the Citrix Knowledge Center (http://support.citrix.com/productdocs/). When you click these links, you will leave the site. We recommend that you book mark this site so you can easily return to it.

Can't find what you're looking for? If you're looking for documentation for previously released versions of this product, go to the Citrix Knowledge Center. For a complete list of links to all product documentation in the Knowledge Center, go to http://support.citrix.com/productdocs/.

Readme

A Readme topic describes the bug fixes and known issues for a particular build of the NetScaler software. Readmes are grouped into two categories, one for maintenance releases and the other for enhancement releases. The following topics provide the consolidated list of bugs fixed and known issues of each release:

- Maintenance Releases
- Enhancement Releases

Maintenance Releases

This section provides the readmes for maintenance releases of NetScaler version 9.3.

- Build 54.4
- Build 53.5
- Build 52.3
- Build 51.5
- Build 50.3

Maintenance Releases

This section provides the readmes for maintenance releases of NetScaler version 9.3.

- Build 54.4
- Build 53.5
- Build 52.3
- Build 51.5
- Build 50.3

Build 54.4

Release version: Citrix(R) NetScaler(R), version 9.3 build 54.4

Replaces build: None

Readme version: 1.0

Release date: December, 2011

Language supported: English (US)

This section describes the changes, bug fixes, and known issues in version 9.3 build 54.4 of the Citrix(R) NetScaler(R), Citrix NetScaler SDX, and Citrix Access Gateway(R) software.

Review the following sections:

- Changes and Fixes
- Known Issues and Workarounds

Note:

- Unless stated otherwise, an issue applies to all build types (Classic, nCore, and nCore VPX) of Citrix NetScaler and Citrix Access Gateway.
- A copy of this readme is also available in the Citrix Knowledge Center at http://support.citrix.com/.

Changes and Fixes

AAA-TM Issues

- Issue ID 93854/0257700: When AAA is in use, and a user who has not already authenticated accesses a URL that contains encoded spaces (%20), after authentication AAA replaces the encoded spaces with the plus (+) character before it attempts to access the URL. When it attempts to redirect to the modified URL, the web server returns a 404 error.

Access Gateway Issues

- Issue ID 93631/0257509: Access Gateway supports client interception by using Intranet Applications. You can configure up to 128 intranet applications in Access Gateway. The previous limit was 32 intranet applications.
- Issue ID 91875/0251005: When users log on to Access Gateway, if there is a delay sending the authentication response, authentication and Access Gateway might fail.
- Issue IDs 93657/0257518 and 93716/0257574: When users attempt to log on two times, Access Gateway detects an active session and starts to initiate the transfer logon process. However, Access Gateway fails to remove the original session and transfer logon fails. If users try to transfer logon again, the logon page appears.
- Issue ID 94181/0258005: If UDP packets from the user device arrive at Access Gateway as multiple packets, Access Gateway truncates the UDP packet if all the truncated fragments fit in one packet and then are sent from the appliance.
- Issue ID 0266865: Access Gateway does not update the MAC address cache for XenApp or XenDesktop when the MAC address of the server changes. User connections fail and you must restart Access Gateway.
- Issue ID 0271589: If you upgrade Access Gateway from Version 9.3 build 51.5 to build 53.5, when users log on successfully with the Access Gateway Plug-in on an iPad, an `"Access Gateway unexpected response"` error appears.

AppFlow Issues

- Issue ID 0274236 (nCore and nCore VPX): When the AppFlow feature is enabled, memory more than the allocated size of buffer may be released. This may result in memory

corruption and packet engine failure.

Application Firewall Issues

- Issue ID 0264504: In some circumstances when a large number of Application Firewall sessions are active, the NetScaler watchdog process can stall and abort the packet engine, causing a system restart.
- Issue ID 0269994: After upgrading a NetScaler appliance that has the Application Firewall enabled and configured from version 9.0 of the NetScaler OS to version 9.3, if the deny URL feature is configured and deny URLs are enabled, memory usage increases significantly and continues to increase over time as denied URLs are accessed.
- Issue ID 0271427: When a user attempts to connect to an Application Firewall-protected web site using an Apple IPhone, the connection would fail with an error.

Cloud Bridge Issues

- Issue ID 90206/0249539 (Classic): The IKE process causes a loop and due to which 100 percent CPU usage is observed.

Configuration Utility Issues

- Issue ID 0269486: In an High availability configuration, the configuration utility does not display the configured route monitors on the NetScaler appliances. Also, when a route monitor is configured to monitor a default route , the configuration utility displayes the secondary node's IP address as 0.0.0.0.

Content Switching Issues

- Issue ID 0264772 (nCore and nCore VPX): The NetScaler appliance does not increment the hit counters that are displayed for content switching policy bindings in the output of the "show cs vserver <name>" command, even though it increments the counters for the total number of policy hits in the output of the "show cs policy [<*policyName*>]" command. The issue occurs with URL-based policies when the “caseSensitive” option for the content switching virtual server is set to “OFF.”

DataStream Issues

- Issue ID 90389/0249698 (nCore and nCore VPX): The MSSQL-ECV monitor uses the default MS SQL protocol version, TDS 7.0, even if you set a different MS SQL protocol version for the monitor.

EdgeSight Monitoring Issues

- Issue ID 0266622 (Classic and nCore): Injection of EdgeSight for NetScaler measurement scripts into the response occurs only if the HTTP content-type header is text/html.

Global Server Load Balancing Issues

- Issue ID 92365/0251432: When a GSLB virtual server that is configured with the static proximity method receives requests that alternately match a separate subset of bound GSLB services, the NetScaler appliance fails to serve each request the GSLB service IP addresses, from the respective subset, in round-robin order. For example, if the GSLB virtual server receives a request R1 that matches services S1, S2, and S3 and a request R2 that matches services S4, S5, and S6, in the order R1, R2, R1, R2, the NetScaler appliance fails to serve R1 the IP addresses of S1, S2, and S3 and R2 the IP addresses of S4, S5, and S6, in round-robin order.

Integrated Caching Issues

- Issue ID 93435/0257335: If a request for an object arrives as the object expires (at the time specified by the "**absExpiry**" parameter), the "**pollEveryTime**" parameter for that object is set to YES. Future requests for the object are sent to the origin server.

Load Balancing Issues

- Issue ID 69918//0233211: Unlike in earlier releases, when you use the “sync gslb config” command or its alias, the “sync config” command, the NetScaler appliance displays a warning that the synchronization of GSLB sites can result in loss of configuration on remote sites, and prompts you to confirm that you want to synchronize the sites. The prompt helps prevent unintentional synchronization that might result from accidental use of the command.

NetScaler SDX Appliance Issues

- Issue ID 0261672: You can download the Management Service build and documentation files, SSL keys and certificates, XVA images, NetScaler instance build and documentation files, and licenses to a local computer as a backup. You can also directly download the technical support file to your local computer and then send it to Citrix support. Earlier you had to use FTP to download these files.

- Issue ID 0268115 (nCore): You cannot change an interface on a NetScaler VPX instance if you have not selected any of the management interfaces (0/1 and 0/2) when provisioning the instance.

  Note: Make sure that the NSVLAN is configured correctly. In case of an incorrect configuration, the instance is not reachable.

- Issue ID 0269055: You can now save the settings of all the NetScaler instances provisioned on the SDX appliance before performing a factory reset. You can use the saved information to reprovision the instances after the reset. For more information, see the SDX Administration Guide at http://support.citrix.com/article/CTX129335.

- Issue ID 0267383 (nCore): You cannot remove a configured NSVLAN by using **"Modify NetScaler Instance"** in the management Service VM user interface.

Networking Issues

- Issue ID 0263530 (nCore and nCore VPX): The NetScaler appliance fails when it processes IPv6 UDP packets for which the appliance has not allocated memory for NAT entry.

SNMP Issues

- Issue ID 93916/0257759: For an SNMP query, the NetScaler appliance returns the value of the SNMP objects svcAvgTransactionTime and svcGrpMemberAvgTransactionTime, in picoseconds.

- Issue ID 0260021 (nCore and nCore VPX): NetScaler appliance returns a different value for the SNMP object **'sslSessionsPerSec'** than the value of the corresponding **'SSL sessions (Rate)'** counter displayed on the Monitoring page.

SSL Issues

- Issue ID 92246/0251327 (Classic): In rare cases, on the NetScaler 12000 appliance, if the combination of a NetScaler response and maximum transmission unit (MTU) is such that the data in the last tcp packet is 8 bytes or less, decryption of data using DES/AES ciphers fails.

System Issues

- Issue ID 89581/0249017: The NetScaler appliance fails due to the internal traffic accessing the buffer.
- Issue ID 90580/0249858: Log records are not generated for the "reboot" command.
- Issue ID 0259201: In rare cases, the appliance restarts when the process monitoring daemon does not recognize the short heartbeat messages from clients.
- Issue ID 0263699: The NetScaler appliance does not process invalid requests that are sent after a connection close.
- Issue ID 88149/0247876: The NetScaler appliance is unable to learn the restarting of the httpd process after the process fails. This causes the NetScaler appliance to drop SYN packets intermittently destined to the NSIP, or MIP, or SNIP address on the appliance.
- Issue ID 93475/0257369: After a data structure,used for tracking NAT info was getting freed to memory, a field related to netbridge configuration, was not getting zeroed out and if same data structure was being picked up again for server side coneection, we tried to send data on the bridge which is not configured so packet was not getting out.
- Issue ID 93586/0257469: The NetScaler appliance forwards Keep Alive probes, from the server, to a client even when the client has advertised a zero window. This causes the client to reset the connection.
- Issue ID 93826/0257673: HTTP requests may acknowledge previous responses that causes packet re-ordering. The NetScaler appliance fails when any configured L7 features, for example rewrite policies, processes these packets.
- Issue ID 94442/0258246 and 93593/0257475: When device name length exceeds 256 characters, then the length stored is truncated. However, the NetScaler appliance allocates more memory to store the device name and while releasing the memory, the appliance releases less memory than the extended. This leads to memory leak.

Known Issues and Workarounds

Access Gateway Issues

- Issue ID 92054/0251163: If you enable multi-stream connection policy settings in either XenApp 6.5 for Windows Server 2008 R2 or XenDestop 5.5 and you install the Access Gateway Plug-in by using an MSI package, Access Gateway does not establish multi-stream connections to the resource, although XenApp and XenDesktop launch on the user device in a single stream.

- Issue ID 91832/0250964: If users logon with the Access Gateway Plug-in and then put the user device into hibernation, when the device resumes from a different network, the Access Gateway Plug-in reconnects, but when users log off, the default route might be deleted. Users can restart their device to obtain the network route.

- Issue IDs 80175/0241433 and 82022/0242906: If you enable split tunneling, split DNS, and assign an intranet IP address on Access Gateway, when users log on with the Access Gateway Plug-in using a mobile broadband wireless device that uses the Sierra driver (for example, Telstra Compass or AT&T USBConnect) on a Windows 7 computer, Domain Name Service (DNS) resolution fails and the home page fails to open. You can use one of the following options to resolve the issue:

    1. Disable split tunneling.

    2. Configure Access Gateway so user connections do not receive an intranet IP address.

    3. Configure the wireless device to use an Ethernet connection instead of a mobile broadband connection. For example:

        a. Disable the setting Windows 7 Mobile Broadband in the Telstra Connection Manager Options dialog box.

        b. Install Sierra Wireless Watcher (6.0.2849) if users connect with an AT&T USBConnect 881 network card. This installs an Ethernet adapter instead of the mobile broadband adapter.

        c. Contact the manufacturer for other devices.

- Issue ID 89427/0248893: If users connect with the Access Gateway Plug-in by using an airtel 3G device and the Repeater Plug-in accelerates VPN traffic after the establishing the VPN connection, when users put the user device in standby or hibernate, when users resume the device, the Access Gateway Plug-in fails to reestablish the connection. When users disable acceleration with the Repeater Plug-in, restoration of the VPN connection is successful. Users can then enable acceleration with the Repeater Plug-in.

- Issue ID 89439/0248898: If users connect with the Access Gateway Plug-in and a T-Mobil 3G device and if you enable split tunneling and assign an intranet IP address to users on

Access Gateway, users cannot connect to either intranet or external resources. To allow users to connect to both internal and external resources, disable split tunneling.

- Issue ID 89791/0249202: If users log on with a Windows-based computer that is not part of a domain by using a 3G network adapter and the Access Gateway Plug-in for Windows, requests that use the host name fail. In this instance, use the fully qualified domain name (FQDN) instead of the host name.

- Issue ID 90675/0249937: If users log on with the Access Gateway Plug-in for Windows and then access a CIFS share by using the Run dialog box, when users navigate to a folder in the share and attempt to copy a file to another file share, users receive an error message and the attempt fails.

- Issue ID 84787/0245136: When you issue the command "sh vpn vserver" on Access Gateway, the number of current ICA connections does not appear when Access Gateway is in Basic mode.

- Issue ID 84986/0245297: If users log on with clientless access and attempt to open an external Web site (such as http://www.google.com) from the Email tab in the Access Interface, users might receive the Access Gateway logon page instead of the external Web site.

- Issue ID 88268/0247968: If users attempt to open a large Microsoft Word file from a Distributed File Share (DFS) hosted on Windows Server 2008 64-bit, the Access Gateway fails.

- Issue ID 81494/0242522: If users access a Distributed File Share on a computer running Windows Server 2008 64-bit, a blank folder appears in the directory path.

- Issue ID 83492/0244134: When users log on using clientless access, a JavaScript error might appear when the logon page opens.

- Issue ID 83819/0244412: If you configure a load balancing virtual server and the destination port is 21, when users log on with the Access Gateway Plug-in, logon is successful but data connections do not go through. When you configure a load balancing virtual server, do not use port 21.

- Issue ID 84894/0245227: When users log off from the Access Gateway Plug-in and then clear the cache in Internet Explorer and Firefox, users might receive an error message that says "`Error. Not a privileged user.`" Access Gateway records an HTTP/1.1 403 Access Forbidden error message in the logs.

- Issue ID 84915/0245243: If users attempt to open and edit a Microsoft Office file from Outlook Web Access, users might receive an error and the file takes a long time to open. To allow users to edit files from Outlook Web Access, do the following:

  1. Create a clientless access Outlook Web Access Profile and enable persistent cookies.

  2. Bind the Outlook Web Outlook regular expression to this profile

  3. Bind the profile so that is assumes the highest priority.
- Issue ID 86122/0246165: If you disable transparent interception and set the force time-out setting, when users log on with the Access Gateway Plug-in for Java, when the time-out period expires, a session time-out message appears on the user device, however the session is not terminated on Access Gateway.

- Issue ID 86123/0246166: If users log on with clientless access in the Firefox Web browser, when users click a link for a virtual application, the tab closes and the application does not start. If users right-click the virtual application and attempt to open it in a new window, the Web Interface appears and users receive the warning "`Published resource shortcuts are currently disabled.`" Users can open the virtual application in Internet Explorer.

- Issue ID 86323/0246328: If you configure single sign-on with Windows and configure the user name with special characters, when users log on to Windows 7 Professional, single sign-on fails. Users receive the error message "`Invalid username or password. Please try again.`" This issue does not occur if users log on to Windows XP.

- Issue IDs 86470/0246469 and 86787/0246736: When users log on with the Access Gateway Plug-in for Windows using Internet Explorer 9, a delay may occur in establishing the connection. The Access Interface, or a custom home page, might take a long time to appear when users log on using Internet Explorer 9.

- Issue ID 86471/0246473: When users log on with the Access Gateway Plug-in by using a Web browser, users might see a delay during logon.

- Issue ID 86722/0246679: When users log on with clientless access using Internet Explorer 9 and connect to SharePoint 2007, some images might not appear correctly.

ACL Issues

- Issue ID 0264933: The NetScaler appliance does not display the correct default values for the icmpType and icmpCode parameters of an extended ACL or ACL6.

Application Firewall Issues

- Issue ID 0259458: Attempts to upload a 30 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.

CloudBridge Issues

- Issue ID 91850/0250982 (nCore and nCore VPX): The NetScaler appliance drops TCP packets when the server has to send a full-size packet, whose DF bit is unset, across the cloud bridge. This happens because of bad checksum.

Command Line Interface Issues

- Issue ID 82908/0243626 (nCore): In certain rare cases, if the NetScaler MPX appliance is subject to conditions of heavy SSL-related traffic, CLI commands fail and report a configuration inconsistency error.

  Workaround: Check for configuration inconsistency by using the "show configstatus" command and reconfigure the appliance under low traffic conditions or during a maintenance period. If this does not resolve the issue, restart the appliance.

DataStream Issues

- Issue ID 83862/0244449 (nCore and nCore VPX): This release does not support IPv6 addresses.

Domain Name System Issues

- Issue ID 93203/0257123 (nCore): A DNS policy that is bound to a GSLB service is not evaluated if the GSLB method is set to dynamic round trip time (RTT).

Load Balancing Issues

- Issue ID 82872/0243593: The setting for maximum requests per connection may be violated during a transaction with the physical server.
- Issue ID 82929/0243645: When using a TCP monitor for a MYSQL service, the MySQL server blocks the MIP for making new connections.
- Issue ID 82996/0243703: The MYSQL monitor shows the service state as UP when no SNIP or MIP is configured.
- Issue ID 86096/0246139: While configuring the WI-EXTENDED monitor, the user will have to provide the value of sitepath in such a way that it does not end with a '/' . For example:

  add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb -dom
- Issue ID 88593/0248222 (nCore): After failover, the '**maxclient**' setting on a service is not honored.
- Issue ID 90271/0249597: Application scripts that parse the servicegroup member name, and use the underscore delimiter (_) to identify fields to be extracted, fail, because the delimiter is now a question mark (?). In NetScaler releases earlier than 9.3, the format is `<service group name>_<IP address>_<port>`. The new format is `<servicegroup name>?<IP address | server name>?<port>`.

- Issue ID 87407/0247289 (nCore and nCore VPX): For an RDP virtual server, the NetScaler appliance automatically maintains persistence through session cookies by using Session Directory, so you need not explicitly configure persistence. Currently, IP address based persistence is not supported, and you cannot disable the implicit session cookie based persistence.

NetScaler SDX Appliance Issues

- Issue ID 265006 (nCore): Tx flow control on the interfaces of a NetScaler VPX instance can cause packets to be dropped instead of transmitted.

  Workaround: Turn off the Tx flow control globally on the interfaces from the management Service VM user interface. On the Configuration tab, in the navigation pane, click System, and then click Interfaces.

- Issue ID 0262505 (nCore): When viewing the built-in or custom reports in the Reporting tab on a NetScaler VPX instance running on the NetScaler SDX 17550/19550/20550/21550 platform, the following message appears: "NO DATA TO CHART".

- Issue ID 86597/0246578 (nCore): Internet Explorer version 9.0 does not load the Configuration tab.

  Workaround: Run Internet Explorer version 9.0 in compatibility mode.

- Issue ID 88515/0248159 (nCore): Client authentication is enabled by default in the Management Service VM. Consequently, HTTPS connections fail when you access the Management Service VM user interface from the Apple Safari browser.

- Issue ID 88556/0248194 (nCore): When provisioning a NetScaler instance, if you have entered invalid NetScaler settings for any of the IP address, Netmask, or Gateway parameters, you cannot modify the values for these parameters.

  Workaround: To correct the parameter values, log on to the NetScaler instance through the Xen Console. You also need to correct the values for this instance in the XenStore. After correcting the values in the both the places, rediscover the NetScaler instances from the Management Service VM user interface without selecting any specific instance, by clicking Rediscovery in the NetScaler Instance pane.

- Issue ID 89148/0248663 (nCore): When you attempt to shut down the SDX appliance from the Management Service VM user interface, the appliance restarts instead of shutting down.

- Issue ID 91605/0250759 (nCore): If the dashboard page on the management service virtual machine is open for more than 4 days, the browser (Internet Explorer 8.0) might report high memory usage.

  Workaround: In the browser, refresh or minimize the page to release the memory.

- Issue ID 0274939: If only a 10/x interface or a 1/x interface is assigned to a NetScaler VPX instance, and the state of that interface is changed from UP to DOWN and then UP again, the instance is not accessible through the NSIP address.

Workaround: Assign a 0/x interface to the VPX instance.

- Issue ID 0275607: In certain cases, if only a 1/x interface is assigned to a NetScaler VPX instance, the instance is unresponsive after it is started.

  Workaround: Assign a 0/x interface to the VPX instance.

NetScaler VPX Appliance Issues

- Issue ID 88057/0247795: If you allocate more than 200MB for caching on a VPX appliance with 2GB RAM or allocate more than 800MB on a VPX appliance with 4GB RAM, memory-intensive features, such as compression and GSLB, stop working.

  Workaround: Reduce the memory allocated for caching.

- Issue ID 94487/0258286: On the Microsoft Hyper-V platform, when there are fragmentation issues on dynamic virtual disks, the NetScaler VPX appliance sends HTTP 5xx responses to requests.

Networking Issues

- Issue ID 0271154 (Classic, nCore, and nCore VPX): The man pages for the commands 'add ns ip','set ns ip','add ns ip6', and 'set ns ip6' displays an incorrect default value for the parameter '**ospfArea**'.

Platform Issues

- Issue ID 0262488 (nCore): On the MPX and SDX 11500/13500/14500/16500/18500/20500 and MPX and SDX 17550/19550/20550/21550 appliances, the network cable does not lock properly into the port and is easy to pull out.

- Issue ID 87419/0247297 (nCore): When you start the remote console from the LOM configuration utility on a NetScaler MPX 11500/13500/14500/16500/18500/20500 or MPX 17550/19550/20550/21550 appliance, remote keyboard redirection does not work.

  Workaround: Reset the LOM firmware. The following error messages might appear on the console during LOM reset, because the LOM module does not respond to the appliance. `ipmi0: KCS error: 01 ipmi0: KCS: Reply address mismatch`

- Issue ID 90018/0249389 (nCore): When you upgrade any MPX appliance, except MPX 15000/17000, restart the appliance, and then apply the default configuration, the 1G interfaces are reset.

- Issue ID 94198/0258025: If a member interface is removed from an LA channel, the LA channel might show negative statistics in the next statistics collection cycle.

Reporting Issues

- Issue ID 85025/0245335 (nCore and nCore VPX): Reporting charts do not support plotting of counters per packet engine.

SSL Issues

- Issue ID 74279/0236509: The cipher TLS1-EXP1024-DES-CBC-SHA is not supported by the NetScaler appliance.
- Issue ID 81850/0242774 (nCore): You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 10G FIPS appliance.

  Workaround: First, decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

  ```
  openssl rsa -in <EncryptedKey.key> > DecryptedKey.out
  ```

- Issue ID 80830/0241961 (nCore): If you attempt to delete an SSL certificate-key pair that is referenced by a certificate revocation list (CRL), the following, incorrect message appears: "`ERROR: Configuration possibly inconsistent. Please check with the 'show configstatus' command or reboot.`" However, the correct message--"`ERROR: Certificate is referenced by a CRL, OCSP responder, virtual server, service, or another certificate`"-- appears upon subsequent attempts to delete the certificate-key pair.
- Issue ID 85393/0245605: A DSA certificate signed with the SHA-2 algorithm is not supported in the client authentication process.

System Issues

- Issue ID 84099/0244639 (nCore): The NetScaler appliance may fail if traffic reaches a load balancing virtual server that uses the token method for load balancing and has connection failover enabled.
- Issue ID 84282/0244774: A global setting of less than 1220 for the maximum segment size (MSS) to use for TCP connections causes an excessive delay in saving the configuration.
- Issue ID 84320/0244792 (nCore and nCore VPX): The NetScaler appliance may fail if a failover happens while high availability (HA) synchronization is in progress.
- Issue ID 94133/0257961: If a server (load balancing virtual server or content switching vserver) is configured with the same IP address, port, and protocol as the server configured in the audit syslog or nslog action, the configured virtual server will be

deleted on upgrading from 9.3_49 or a prior build to a build later than 9.3_49.

Workaround: Do the following:

1. Remove the audit policy and action.
2. Add the deleted virtual server.
3. Add the audit policy and action.
4. Save the configuration.

Web Interface Issues

- Issue ID 89052/0248592 (nCore and nCore VPX): The response from a Web Interface site that is configured in direct mode may have Java errors.

XML Issues

- Issue ID 81650/0242628: The NetScaler import utility validates XML schemas during import, but it may fail to validate certain XHTML files being imported as XMLSchema. These invalid XMLSchema's are rejected if the user tries to use them in profile configuration (XMLValidation binding).
- Issue ID 82058/0242928: The 'unique' element in the XML schema is currently not supported.
- Issue ID 82059/0242929: The 'redefine' element in the XML schema is currently not supported.
- Issue ID 82069/0242939: When the Application Firewall validates XML messages, it does not validate the contents of elements that are defined as type "any" in the applicable XML schema. Specifically, it treats these elements as if the processContent attribute was set to "skip".

  Workaround: Replace the "any" type definitions in the XML schemas with definitions of the actual elements that occur in the XML message. (The "any" type is rarely used.)

XML API Issues

- Issue ID 80170/0241429: The syntax of the "unset servicegroup" command has been changed to allow unsetting the parameters of the service group members. This can cause XML API incompatibility with respect to the "unset servicegroup" command.

Build 53.5

Release version: Citrix(R) NetScaler(R), version 9.3 build 53.5

Replaces build: None

Readme version: 1.0

Release date: November 2011

Language supported: English (US)

This section describes the changes, bug fixes, and known issues in version 9.3, build 53.5 of the Citrix(R) NetScaler(R), Citrix NetScaler SDX, and Citrix Access Gateway(TM) software.

Review the following sections:

- Changes and Fixes
- Known Issues and Workarounds

Note:

- Unless stated otherwise, an issue applies to all build types (Classic, nCore, and nCore VPX) of Citrix NetScaler and Citrix Access Gateway.
- A copy of this readme is also available in the Citrix Knowledge Center at http://support.citrix.com/.

Changes and Fixes

AAA-TM Issues

- Issue IDs 93635/0257512, 0263061, and 0263473: The NetScaler appliance fails in the following scenario:

  1. A user uses invalid credentials to log on to a AAA-TM authentication virtual server, and then sends the virtual server a second request.

  2. The user's browser reuses the TCP connection for the second request.

Access Gateway Issues

- Issue ID 91344/0250516: The multi-stream ICA feature allows you to partition multiple ICA streams in the same session. With multi-stream ICA, you can partition a single TCP connection into multiple streams based on different types of traffic that are typical for session reliability.

- Issue ID 91453/0250615: Occasionally, when users log on with the Access Gateway Plug-in, and a Web browser sends a resources.js request that contains a session cookie (NSC_AAAC), Access Gateway proxies the request to the server and returns an HTTP 404 Not Found error. Unnecessarily, the user receives an NTLM authentication message prompting them to enter credentials.

- Issue ID 92124/0251222: If you configure bookmarks on Access Gateway as a reverse proxy and users do not connect with clientless access, Access Gateway might fail.

- Issue ID 93384/0257288: You can create a session or preauthentication policy to check for REG_MULTI_NZ and REG_BINARY registry types on the user device.

- Issue ID 94534/0258329: When you configure two appliances as a high availability pair, if you create a session policy with a name longer than 31 characters, when users are logged on with the Access Gateway Plug-in,if the primary appliance becomes unavailable, failover to the secondary appliance does not occur and the connection fails.

AppFlow Issues

- Issue IDs 94685/0258921 and 0264518 (nCore and nCore VPX): The NetScaler appliance may crash if you try to delete an AppFlow policy or action while traffic affected by the

policy or action is flowing through the appliance.

Application Firewall Issues

- Issue ID 83695/0244305: On nCore systems with the application firewall enabled, the following counters may show incorrect values:
  - (CLI) Opening client connections
  - (CLI) Established client connections
  - (GUI) Opening connections
  - (GUI) Established connections
  - (GUI) Active Server connections
- Issue ID 92641/0251685 (nCore and nCore VPX): The amount of memory available to the Application Firewall is much lower than the amount of memory available to the NetScaler appliance.
- Issue ID 93940/0257771: When both AAA SSO and the Application Firewall are enabled on the NetScaler appliance, and an advanced Application Firewall profile is bound to global or a bind point, the appliance sends incorrect HTTP POST requests to the web server, breaking web server functionality.

Cache Redirection Issues

- Issue ID 92791/0251815: The NetScaler appliance fails in the following scenario:
  1. A cache redirection virtual server is configured with a listen policy, RNAT is configured, and TCP proxy is enabled for RNAT (by using the "set rnatparam -tcpproxy ENABLED" command).
  2. A client sends the appliance a request meant for the origin server. The request satisfies the RNAT criteria but does not match the listen policy that is configured for the cache redirection virtual server.
  3. Another client sends the appliance a request for the same origin server and, this time, the request matches the listen policy that is configured for the cache redirection virtual server.

Command Line Interface Issues

- Issue ID 92966/0251975: If you are using the Perl package Net::SSH::Perl, the NetScaler appliance may not allow new connections after the limit for maximum number of connections has been reached, even if some users have logged out. Make sure that your

Perl script has the following line: $ssh->login("$user", "$password", 1); instead of: $ssh->login("$user", "$password");

Configuration Utility Issues

- Issue ID 82499/0243259: In NetScaler release 9.3, the PHP version has been upgraded from 5.2.6 to 5.3.8.
- Issue ID 92919/0251932: In the "Select the Template Type" dialog box (AppExpert > Templates > Entity Templates > Add), the option for creating a load balancing virtual server template is not available, and the "SSL Vserver" option is listed more than once.

DataStream Issues

- Issue ID 92394/0251460 (nCore and nCore VPX): If a client cancels an SQL query before the server responds to a query it sent earlier, load balancing fails for subsequent queries sent by the client. The NetScaler appliance forwards all subsequent queries sent by that client to the same database server.

Content Switching Issues

- Issue ID 93339/0257249 (nCore and nCore VPX): A content switching virtual server does not serve a client request in the following scenario:
  - Three or more advanced policies that use the "MATCHES_LOCATION(<location>)" function are bound to the content switching virtual server.
  - The source IP address of the request does not match any location in the location database.

Integrated Caching Issues

- Issue ID 94009/0257849: The NetScaler appliance fails if the maximum response size (maxResSize) of a single object in the integrated cache exceeds 100 MB.
- Issue ID 94708/0258932: The NetScaler appliance does not retransmit data for a 304 Not modified cache hit when RTO (round trip timeout) is hit.

Load Balancing Issues

- Issue ID 92390/0251455: Data transfer might stop after a failover in the following scenario: Stateful connection failover is enabled on the load balancing virtual server that is managing the connection, and the failover was immediately preceded by a burst of traffic.

- Issue ID 87201/0247100 (Classic and nCore): If a load balancing virtual server for TCP services has stateful connection failover enabled, an established connection may be broken if a failover occurs more than once while a large amount of data is being transferred.

NetScaler SDX Appliance Issues

- Issue ID 92664/0251706 (nCore): You can now perform the following actions:

  - Provision a NetScaler VPX instance on a subnet that is different from the subnet of the management service VM. Traffic between the management service VM and the NetScaler VPX instance is routed.
  - Specify that only secure communication is allowed between the management service VM and the NetScaler instances.

  - Specify that the SDX appliance can be accessed only over a secure channel (https instead of http).

  - Apply administrative configuration on a NetScaler instance at a later time if the instance is not reachable from the management service VM.
- Issue ID 0261338 (nCore): If you create a NetScaler IP address from the NetScaler Configuration node in the Configuration tab of the management Service VM user interface, the default IP address created is a subnet IP (SNIP) address and not the mapped IP (MIP) address.

Rewrite Issues

- Issue ID 92586/0251637: Rewrite policies are not applied to HTTP requests that use the HTTP method CONNECT.

- Issue ID 92739/0251770: When an HTTP message body is extremely large (2 GB or more), the replace_all, insert_all, insert_before, and insert_after_all rewrite actions cause the NetScaler to crash.

SNMP Issues

- Issue ID 93003/0252009: A new SNMP OID, sysStatisticsTime (1.3.6.1.4.1.5951.4.1.1.41.17), returns the interval at which various statistical counters are updated.

- Issue ID 93469/0257366: If a content switching classic policy is rebound to the content switching virtual server, an SNMPWALK operation on the csPolicyHits SNMP object returns an error.

SSL Issues

- Issue ID 91408/0250575 (nCore VPX): If OCSP check is enabled on a NetScaler VPX appliance, and the appliance receives the client key exchange and client certificate as part of a single record, the SSL handshake fails.
- Issue IDs 93373/0257280, 0264151, and 0264043 (nCore): If there is a delay between HA health monitoring and SSL card monitoring, HA health monitoring reports that the SSL card is DOWN.

System Issues

- Issue IDs 89829/0249234 and 93515/0257405: The "diff ns config" CLI command erroneously audits a command more than once and displays the error message "Already audited command" in its output. This issue is observed when the command attribute that is treated as the unique ID for the command occurs in multiple records in the NetScaler database. For example, the host name that you specify in the "add dns nsRec" command is treated as the unique ID for the command when a database record is created. If the "add dns nsRec" command is used to assign multiple IP addresses to a host, the host name can occur in multiple records and, consequently, lead to multiple audits of the "add dns nsRec" command when you use "diff ns config."

  Note: As a result of the changes that were made to resolve this issue, the "mx" parameter, which is required in the "add dns mxRec" and "set dns mxRec" commands, is now also required in the "unset dns mxRec" command. The syntax of the "unset dns mxRec" command has changed as follows:

  Before: unset dns mxRec <domain> -TTL

  After: unset dns mxRec <domain> -mx <string> -TTL

- Issue ID 92046/0251155: If a NetScaler appliance is unable to determine the link status of an interface and stores an invalid link-status value in the internal database, the appliance fails.
- Issue ID 94767/0258978: If you use the configuration utility to delete all the configured NTP servers, configurations in one of the startup scripts (rc.netscaler) file are lost.

Known Issues and Workarounds

Access Gateway Issues

- Issue ID 92054/0251163: If you enable multi-stream connection policy settings in either XenApp 6.5 for Windows Server 2008 R2 or XenDestop 5.5 and you install the Access Gateway Plug-in by using an MSI package, Access Gateway does not establish multi-stream connections to the resource, although XenApp and XenDesktop launch on the user device in a single stream.

- Issue ID 91832/0250964: If users logon with the Access Gateway Plug-in and then put the user device into hibernation, when the device resumes from a different network, the Access Gateway Plug-in reconnects, but when users log off, the default route might be deleted. Users can restart their device to obtain the network route.

- Issue IDs 80175/0241433 and 82022/0242906: If you enable split tunneling, split DNS, and assign an intranet IP address on Access Gateway, when users log on with the Access Gateway Plug-in using a mobile broadband wireless device that uses the Sierra driver (for example, Telstra Compass or AT&T USBConnect) on a Windows 7 computer, Domain Name Service (DNS) resolution fails and the home page fails to open. You can use one of the following options to resolve the issue:

    - Disable split tunneling

    - Configure Access Gateway so user connections do not receive an intranet IP address.

    - Configure the wireless device to use an Ethernet connection instead of a mobile broadband connection. For example:

        - Disable the setting Windows 7 Mobile Broadband in the Telstra Connection Manager Options dialog box.

        - Install Sierra Wireless Watcher (6.0.2849) if users connect with an AT&T USBConnect 881 network card. This installs an Ethernet adapter instead of the mobile broadband adapter.

        - Contact the manufacturer for other devices.
- Issue ID 89427/0248893: If users connect with the Access Gateway Plug-in by using an airtel 3G device and the Repeater Plug-in accelerates VPN traffic after the establishing the VPN connection, when users put the user device in standby or hibernate, when users resume the device, the Access Gateway Plug-in fails to reestablish the connection. When users disable acceleration with the Repeater Plug-in, restoration of the VPN connection is successful. Users can then enable acceleration with the Repeater Plug-in.

- Issue ID 89439/0248898: If users connect with the Access Gateway Plug-in and a T-Mobil 3G device and if you enable split tunneling and assign an intranet IP address to users on

Access Gateway, users cannot connect to either intranet or external resources. To allow users to connect to both internal and external resources, disable split tunneling.

- Issue ID 89791/0249202: If users log on with a Windows-based computer that is not part of a domain by using a 3G network adapter and the Access Gateway Plug-in for Windows, requests that use the host name fail. In this instance, use the fully qualified domain name (FQDN) instead of the host name.

- Issue ID 90675/0249937: If users log on with the Access Gateway Plug-in for Windows and then access a CIFS share by using the Run dialog box, when users navigate to a folder in the share and attempt to copy a file to another file share, users receive an error message and the attempt fails.

- Issue ID 84787/0245136: When you issue the command "sh vpn vserver" on Access Gateway, the number of current ICA connections does not appear when Access Gateway is in Basic mode.

- Issue ID 84986/0245297: If users log on with clientless access and attempt to open an external Web site (such as http://www.google.com) from the Email tab in the Access Interface, users might receive the Access Gateway logon page instead of the external Web site.

- Issue ID 88268/0247968: If users attempt to open a large Microsoft Word file from a Distributed File Share (DFS) hosted on Windows Server 2008 64-bit, the Access Gateway fails.

- Issue ID 81494/0242522: If users access a Distributed File Share on a computer running Windows Server 2008 64-bit, a blank folder appears in the directory path.

- Issue ID 83492/0244134: When users log on using clientless access, a JavaScript error might appear when the logon page opens.

- Issue ID 83819/0244412: If you configure a load balancing virtual server and the destination port is 21, when users log on with the Access Gateway Plug-in, logon is successful but data connections do not go through. When you configure a load balancing virtual server, do not use port 21.

- Issue ID 84894/0245227: When users log off from the Access Gateway Plug-in and then clear the cache in Internet Explorer and Firefox, users might receive an error message that says "Error. Not a privileged user." Access Gateway records an HTTP/1.1 403 Access Forbidden error message in the logs.

- Issue ID 84915/0245243: If users attempt to open and edit a Microsoft Office file from Outlook Web Access, users might receive an error and the file takes a long time to open. To allow users to edit files from Outlook Web Access, do the following:

  1. Create a clientless access Outlook Web Access Profile and enable persistent cookies.

  2. Bind the Outlook Web Outlook regular expression to this profile.

  3. Bind the profile so that is assumes the highest priority.
- Issue ID 86122/0246165: If you disable transparent interception and set the force time-out setting, when users log on with the Access Gateway Plug-in for Java, when the time-out period expires, a session time-out message appears on the user device, however the session is not terminated on Access Gateway.

- Issue ID 86123/0246166: If users log on with clientless access in the Firefox Web browser, when users click a link for a virtual application, the tab closes and the application does not start. If users right-click the virtual application and attempt to open it in a new window, the Web Interface appears and users receive the warning "Published resource shortcuts are currently disabled." Users can open the virtual application in Internet Explorer.

- Issue ID 86323/0246328: If you configure single sign-on with Windows and configure the user name with special characters, when users log on to Windows 7 Professional, single sign-on fails. Users receive the error message "Invalid username or password. Please try again." This issue does not occur if users log on to Windows XP.

- Issue IDs 86470/0246469 and 86787/0246736: When users log on with the Access Gateway Plug-in for Windows using Internet Explorer 9, a delay may occur in establishing the connection. The Access Interface, or a custom home page, might take a long time to appear when users log on using Internet Explorer 9.

- Issue ID 86471/0246473: When users log on with the Access Gateway Plug-in by using a Web browser, users might see a delay during logon.

- Issue ID 86722/0246679: When users log on with clientless access using Internet Explorer 9 and connect to SharePoint 2007, some images might not appear correctly.

ACL Issues

- Issue ID 0264933: The NetScaler appliance does not display the correct default values for the icmpType and icmp Code parameters of an extended ACL or ACL6.

Application Firewall Issues

- Issue ID 83089/0243784: The users can look at the default signature rules in configuration utility, but it will be useful to have a comprehensive list of all the rules accessible in documentation or white papers for users to review.

- Issue ID 0259458: Attempts to upload a 30 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.

CloudBridge Issues

- Issue ID 91850/0250982 (nCore and nCore VPX): The NetScaler appliance drops TCP packets when the server has to send a full-size packet, whose DF bit is unset, across the cloud bridge. This happens because of bad checksum.

Command Line Interface Issues

- Issue ID 82908/0243626 (nCore): In certain rare cases, if the NetScaler MPX appliance is subject to conditions of heavy SSL-related traffic, CLI commands fail and report a configuration inconsistency error.

  Workaround: Check for configuration inconsistency by using the "show configstatus" command and reconfigure the appliance under low traffic conditions or during a maintenance period. If this does not resolve the issue, restart the appliance.

DataStream Issues

- Issue ID 83862/0244449 (nCore and nCore VPX): This release does not support IPv6 addresses.

Domain Name System Issues

- Issue ID 93203/0257123 (nCore): A DNS policy that is bound to a GSLB service is not evaluated if the GSLB method is set to dynamic round trip time (RTT).

Load Balancing Issues

- Issue ID 82872/0243593: The setting for maximum requests per connection may be violated during a transaction with the physical server.
- Issue ID 82929/0243645: When using a TCP monitor for a MYSQL service, the MySQL server blocks the MIP for making new connections.
- Issue ID 82996/0243703: The MYSQL monitor shows the service state as UP when no SNIP or MIP is configured.
- Issue ID 86096/0246139: While configuring the WI-EXTENDED monitor, the user will have to provide the value of sitepath in such a way that it does not end with a '/' . For example: add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb -domain ccc
- Issue ID 87407/0247289 (nCore and nCore VPX): When an RDP service is configured, the NetScaler appliance automatically maintains persistence through session cookies using Session Directory. You need not explicitly configure persistency on NetScaler. In some situations, (where multiple persons use same user-login credentials) session cookie persistence may not be helpful, and IP-based persistence methods are necessary. Future releases will support IP address based persistency. In some other situations, load balancing of the RDP services without persistence may be necessary. That is, each new

connection to an RDP virtual server needs to be load balanced irrespective of a user's disconnected session existing on a terminal server.

- Issue ID 88593/0248222 (nCore): After failover, the "maxclient" setting on a service is not honored.

- Issue ID 90271/0249597: Application scripts that parse the servicegroup member name, and use the underscore delimiter (_) to identify fields to be extracted, fail, because the delimiter is now a question mark (?). In NetScaler releases earlier than 9.3, the format is <service group name>_<IP address>_<port>. The new format is <servicegroup name>?<IP address | server name>?<port>.

NetScaler SDX Appliance Issues

- Issue ID 86597/0246578 (nCore): Internet Explorer version 9.0 does not load the Configuration tab.

  Workaround: Run Internet Explorer version 9.0 in compatibility mode.

- Issue ID 88515/0248159 (nCore): Client authentication is enabled by default in the Management Service VM. Consequently, HTTPS connections fail when you access the Management Service VM user interface from the Apple Safari browser.

- Issue ID 88556/0248194 (nCore): When provisioning a NetScaler instance, if you have entered invalid NetScaler settings for any of the IP address, Netmask, or Gateway parameters, you cannot modify the values for these parameters.

  Workaround: To correct the parameter values, log on to the NetScaler instance through the Xen Console. You also need to correct the values for this instance in the XenStore. After correcting the values in the both the places, rediscover the NetScaler instances from the Management Service VM user interface without selecting any specific instance, by clicking Rediscovery in the NetScaler Instance pane.

- Issue ID 89148/0248663 (nCore): When you attempt to shut down the SDX appliance from the Management Service VM user interface, the appliance restarts instead of shutting down.

- Issue ID 91605/0250759 (nCore): If the dashboard page on the management service virtual machine is open for more than 4 days, the browser (Internet Explorer 8.0) might report high memory usage.

  Workaround: In the browser, refresh or minimize the page to release the memory.

- Issue ID 0262505 (nCore): When viewing the built-in or custom reports in the Reporting tab on a NetScaler VPX instance running on the NetScaler SDX 17550/19550/20550/21550 platform, the following message appears: "NO DATA TO CHART".

- Issue ID 0265006 (nCore): Tx flow control on the interfaces of a NetScaler VPX instance can cause packets to be dropped instead of transmitted.

Workaround: Turn off the Tx flow control globally on the interfaces from the management Service VM user interface. On the Configuration tab, in the navigation pane, click System, and then click Interfaces.

- Issue ID 0268115 (nCore): You cannot change an interface on a NetScaler VPX instance if you have not selected any of the management interfaces (0/1 and 0/2) when provisioning the instance.

  Workaround: First, add an interface to the NetScaler VPX instance. After the instance restarts, remove the interfaces that are not required.

  Note: Make sure that the NSVLAN is configured correctly. In case of an incorrect configuration, the instance is not reachable.

NetScaler VPX Appliance Issues

- Issue ID 88057/0247795: If you allocate more than 200MB for caching on a VPX appliance with 2GB RAM or allocate more than 800MB on a VPX appliance with 4GB RAM, memory-intensive features, such as compression and GSLB, stop working.

  Workaround: Reduce the memory allocated for caching.

Platform Issues

- Issue ID 87419/0247297 (nCore): When you start the remote console from the LOM configuration utility on a NetScaler MPX 11500/13500/14500/16500/18500/20500 or MPX 17550/19550/20550/21550 appliance, remote keyboard redirection does not work.

  Workaround: Reset the LOM firmware. The following error messages might appear on the console during LOM reset, because the LOM module does not respond to the appliance. ipmi0: KCS error: 01 ipmi0: KCS: Reply address mismatch

- Issue ID 90018/0249389 (nCore): When you upgrade any MPX appliance, except MPX 15000/17000, restart the appliance, and then apply the default configuration, the 1G interfaces are reset.

- Issue ID 94198/0258025: If a member interface is removed from an LA channel, the LA channel might show negative statistics in the next statistics collection cycle.

- Issue ID 0262488 (nCore): On the MPX and SDX 11500/13500/14500/16500/18500/20500 and MPX and SDX 17550/19550/20550/21550 appliances, the network cable does not lock properly into the port and is easy to pull out.

Reporting Issues

- Issue ID 85025/0245335 (nCore and nCore VPX): Reporting charts do not support plotting of counters per packet engine

SSL Issues

- Issue ID 74279/0236509: The cipher TLS1-EXP1024-DES-CBC-SHA is not supported by the NetScaler appliance.
- Issue ID 80830/0241961 (nCore): If you attempt to delete an SSL certificate-key pair that is referenced by a certificate revocation list (CRL), the following, incorrect message appears: "ERROR: Configuration possibly inconsistent. Please check with the 'show configstatus' command or reboot." However, the correct message--"ERROR: Certificate is referenced by a CRL, OCSP responder, virtual server, service, or another certificate"-- appears upon subsequent attempts to delete the certificate-key pair.
- Issue ID 81850/0242774 (nCore): You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 10G FIPS appliance.

  Workaround: First, decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

  openssl rsa -in <EncryptedKey.key> > DecryptedKey.out
- Issue ID 85393/0245605: A DSA certificate signed with the SHA-2 algorithm is not supported in the client authentication process.

System Issues

- Issue ID 84099/0244639 (nCore): The NetScaler appliance may fail if traffic reaches a load balancing virtual server that uses the token method for load balancing and has connection failover enabled.
- Issue ID 84282/0244774: A global setting of less than 1220 for the maximum segment size (MSS) to use for TCP connections causes an excessive delay in saving the configuration.
- Issue ID 84320/0244792 (nCore and nCore VPX): The NetScaler appliance may fail if failover happens while high availability (HA) synchronization is in progress.
- Issue ID 94133/0257961: If a server (lb virtual server or cs vserver) is configured with the same IP address, port, and protocol as the server configured in the audit syslog or nslog action, the configured virtual server will be deleted on upgrading from 9.3_47.5 to 9.3_51.5.

  Workaround: Do the following:

  1. Remove the audit policy and action

2. Add the deleted virtual server
3. Add the audit policy and action
4. Save the configuration

Web Interface Issues

- Issue ID 89052/0248592 (nCore and nCore VPX): The response from a Web Interface site that is configured in direct mode may have Java errors.

XML Issues

- Issue ID 81650/0242628: The NetScaler import utility validates XML schemas during import, but it may fail to validate certain XHTML files being imported as XMLSchema. These invalid XMLSchema's are rejected if the user tries to use them in profile configuration (XMLValidation binding).
- Issue ID 82058/0242928: The "unique" element in the XML schema is currently not supported.
- Issue ID 82059/0242929: The "redefine" element in the XML schema is currently not supported.
- Issue ID 82069/0242939: When the Application Firewall validates XML messages, it does not validate the contents of elements that are defined as type "any" in the applicable XML schema. Specifically, it treats these elements as if the processContent attribute was set to "skip".

  Workaround: Replace the "any" type definitions in the XML schemas with definitions of the actual elements that occur in the XML message. (The "any" type is rarely used.)
- Issue ID 83707/0244316: The Import feature for Schema and WSDL files does not support Non-ASCII characters. If an importing WSDL/Schema file contains Non-ASCII characters then it results in a partial import.

  Workaround: Convert XML schema or WSDL files to ASCII before importing them.

XML API Issues

- Issue ID 80170/0241429: The syntax of the "unset servicegroup" command has been changed to allow unsetting the parameters of the service group members. This change can cause XML API incompatibility with respect to the "unset servicegroup" command.

Build 52.3

Release version: Citrix(R) NetScaler(R), version 9.3 build 52.3

Replaces build: None

Release date: October 2011

Language supported: English (US)

Readme version: 2.0

This readme is applicable for version 9.3 build 52.3 of the Citrix(R) NetScaler(R), Citrix NetScaler SDX, and Citrix Access Gateway(TM) software.

Review the following sections:

Note:

- Unless stated otherwise, this readme applies to all build types (Classic, nCore, and nCore VPX) of Citrix NetScaler and Citrix Access Gateway.
- A copy of this readme is also available in the Citrix Knowledge Center at http://support.citrix.com/.

Changes and Fixes

Access Gateway Issues

- Issue ID 86965: When users connect to Access Gateway and you have single sign-on enabled, when users connect to a resource, occasionally Access Gateway returns a 500 Internal Server Error when the resource does not accept a 0 content length during NTLM authentication.
- Issue ID 88409: If 16 users are changing their passwords simultaneously, new users cannot log on.
- Issue ID 91811: If you enable client choices and users can log on with the Access Gateway Plug-in or by using clientless access, when users try to transfer files, an Internal Server Error 29 appears.
- Issue ID 91285: When two Access Gateway appliances are configured as part of a high availability pair and you configure double-source authentication, occasionally if the primary node fails, the secondary node also fails to accept connections and Access Gateway subsequently fails.

Application Firewall Issues

- Issue ID 83366: When the application firewall is configured, the NetScaler appliance may fail because of an issue in delivering the learning messages to the "aslearn" daemon.
- Issue ID 86782: Importing appfw objects such as signatures, htmlerror page, and so on, may fail on first attempt if the import command contains a very long source URL. The command will succeed if executed a second time.
- Issue ID 91899: The Application Firewall fails when a web page with a significantly large number of unique URLs is processed.

Configuration Utility Issues

- Issue 93321: If, when adding a new TCP profile, you specify a value greater than 128 for the "Maximum Burst Limit" field, the appliance adds a minus symbol (-) before the value.

- Issue ID 93358 (Classic and nCore): When you install a CA certificate on a NetScaler FIPS appliance by using the configuration utility, FIPS Key Name is not a required parameter.

Domain Name System Issues

- Issue ID 92383: If a DNS query made by the NetScaler appliance leads to a chain of CNAME responses and one of the intermediate CNAME response expires before the other responses because it has the least time-to-live (TTL) value, the bailiwick check is not handled correctly during subsequent attempts to resolve the same DNS query. Consequently, subsequent attempts fail intermittently. Additionally, when the appliance is functioning as an end resolver, if the length of the domain name in the query is less than 64 characters and the domain name in an intermediate CNAME response is greater than 64 characters, CNAME references are not handled correctly. Consequently, the appliance fails when you run the "flush dns proxyRecords" command.

Global Server Load Balancing Issues

- Issue ID 85912: In a large GSLB configuration, if persistence session exchange is enabled between sites and the configuration of GSLB virtual servers is not symmetric across the participating sites, then the NetScaler appliance may fail.
- Issue ID 87244 (nCore and nCore VPX): A NetScaler appliance running release 9.2 fails under the following sequence of events:

  1. GSLB is configured on the appliance, and it receives a metrics exchange protocol (MEP) connection from another GSLB site that has a lower IP address and is running a NetScaler release earlier than 9.2.
  2. The MEP connection is received by a non-owner core and the state of the MEP connection flaps (goes down and comes back up within an interval of about 10 seconds).
  3. After the MEP connection flaps, the GSLB site running NetScaler 9.2 attempts to share persistence sessions with the GSLB site running the pre-9.2 release.

Integrated Caching Issues

- Issue ID 90335: The "show cache object" CLI command can corrupt NetScaler memory, causing the appliance to fail.

Load Balancing Issues

- Issue ID 91869: The counter that indicates the number of connections made to a load balancing virtual server is not incremented correctly when the load balancing virtual server is bound to a content switching virtual server.

Networking Issues

- Issue ID 87163: If a response packet is dropped during high availability failover, the NetScaler appliance does not block fragmented request packets.
- Issue ID 91703: On failover, the bgp sessions on the secondary NetScaler in a HA-INC pair are reset. This is applicable to cl, nCore and vpx builds.
- Issue ID 91886 (nCore and nCore VPX): If the NetScaler appliance receives a monitoring packet that it sent out for monitoring a service, the appliance may fail to send further monitoring probes for the service.
- Issue ID 92773: The NetScaler appliance fails if a client tries to establish an active FTP session with a server that is reachable through a Link Load Balancing (LLB) route.

NITRO API Issues

- Issue ID 93938: The data type for the "statechangetimeseconds" field for the lbvserver and csvserver classes is not consistent. The lbvserver class uses the integer data type and the csvserver class uses the date data type.

Platform Issues

- Issue ID 92406 (Classic and nCore): On the NetScaler MPX 7500/9500 and MPX 9700/10500/12500/15500 appliances with an Intel 1GB interface, the health monitoring system automatically performs a warm restart if the appliance does not respond to health checks. If the appliance continues to be unresponsive, you have to perform a hard reboot.
- Issue IDs 92435 and 92677 (nCore): In certain rare conditions, access to the BMC device for health check data may fail on the MPX 11500/13500/14500/16500/18500 platform. Consequently, the load on the management CPU significantly escalates and the appliance becomes unresponsive.

Policies Issues

- Issue ID 92982: The NetScaler appliance fails when the rewrite action "replace_all" is applied to a response that is being generated by a content filter action.

Rewrite Issues

- Issue ID 88938: The NetScaler appliance may sometimes fail to rewrite HTTP responses that use chunked encoding.
- Issue ID 92968: If a rewrite policy with the "log" action specified is bound to a TCP virtual server, when the policy matches a connection, the NetScaler appliance fails.

SDX Appliance Issues

- Issue ID 92408: If the Management Service VM is in an idle state, the database connection sometimes times out internally causing continous login failures.

SNMP Issues

- Issue ID 92172: Some types of incorrectly formatted SNMP OID requests may lead to failure of the NetScaler appliance.

System Issues

- Issue ID 91005: Client and server-side connections for RNAT do not log TCP 4 tuple information, connection duration, bytes transferred, and connection duration.
- Issue ID 91561: If you run the shell command showtechsupport to create a collector file, "pciconf -lcv" is also executed. The output of the command appears under <collectorfile>/shell/pciconf-lcv.out.
- Issue ID 91606 (nCore): When a layout file is used with nCore to run PEs on different CPUs, if a PE ID is greater than the ID of the CPU running the PE, the profiler fails.
- Issue ID 91693: If there is no URI-QUERY string after the "?" (example: http://search.citrix.com/search?), the client side weblog report must indicate the NULL value by showing "-". However, nothing is recorded in the weblog report.
- Issue ID 92654: Setting the deprecated "recvbufsize" parameter through the "set ns tcpparam" command throws an "argument deprecated" error in the CLI but still modifies the global tcpprofile nstcp_default_profile.
- Issue IDs 93094 and 93983: If the nsconfigaudit tool cannot allocate the memory required for comparing large configurations, the tool fails.

- Issue ID 94674: If a server (lb virtual server or cs vserver) is configured with the same IP address, port, and protocol as the server configured in the audit syslog or nslog action, the configured virtual server will be deleted on reboot.

Web Interface Issues

- Issue ID 92859 (nCore and nCore VPX): The "Enable access through mobile receiver" option in the Web Interface GUI wizard activates web interface sites for most mobile platforms but is known to work for the following:
  - iPhone Receiver.
  - iPad Receiver.
  - Android Receiver.
  - Blackberry Receiver.
  - Mac Receiver .
  - iPad web browser.
  - Wyse Terminals.

Known Issues and Workarounds

Access Gateway Issues

- Issue ID 92054: If you enable multi-stream connection policy settings in either XenApp 6.5 for Windows Server 2008 R2 or XenDesktop 5.5 and you install the Access Gateway Plug-in by using an MSI package, Access Gateway does not establish multi-stream connections to the resource, although XenApp and XenDesktop launch on the user device in a single stream.

- Issue ID 91832: If users logon with the Access Gateway Plug-in and then put the user device into hibernation, when the device resumes from a different network, the Access Gateway Plug-in reconnects, but when users log off, the default route might be deleted. Users can restart their device to obtain the network route.

- Issue IDs 80175 and 82022: If you enable split tunneling, split DNS, and assign an intranet IP address on Access Gateway, when users log on with the Access Gateway Plug-in using a mobile broadband wireless device that uses the Sierra driver (for example, Telstra Compass or AT&T USBConnect) on a Windows 7 computer, Domain Name Service (DNS) resolution fails and the home page fails to open. You can use one of the following options to resolve the issue:

    1. Disable split tunneling.

    2. Configure Access Gateway so user connections do not receive an intranet IP address.

    3. Configure the wireless device to use an Ethernet connection instead of a mobile broadband connection. For example:

        - Disable the setting Windows 7 Mobile Broadband in the Telstra Connection Manager Options dialog box.

        - Install Sierra Wireless Watcher (6.0.2849) if users connect with an AT&T USBConnect 881 network card. This installs an Ethernet adapter instead of the mobile broadband adapter.

        - Contact the manufacturer for other devices.
- Issue ID 89427: If users connect with the Access Gateway Plug-in by using an airtel 3G device and the Repeater Plug-in accelerates VPN traffic after the establishing the VPN connection, when users put the user device in standby or hibernate, when users resume the device, the Access Gateway Plug-in fails to reestablish the connection. When users disable acceleration with the Repeater Plug-in, restoration of the VPN connection is successful. Users can then enable acceleration with the Repeater Plug-in.

- Issue ID 89439: If users connect with the Access Gateway Plug-in and a T-Mobil 3G device and if you enable split tunneling and assign an intranet IP address to users on Access Gateway, users cannot connect to either intranet or external resources. To allow

users to connect to both internal and external resources, disable split tunneling.

- Issue ID 89791: If users log on with a Windows-based computer that is not part of a domain by using a 3G network adapter and the Access Gateway Plug-in for Windows, requests that use the host name fail. In this instance, use the fully qualified domain name (FQDN) instead of the host name.

- Issue ID 90675: If users log on with the Access Gateway Plug-in for Windows and then access a CIFS share by using the Run dialog box, when users navigate to a folder in the share and attempt to copy a file to another file share, users receive an error message and the attempt fails.

- Issue ID 84787: When you issue the command "sh vpn vserver" on Access Gateway, the number of current ICA connections does not appear when Access Gateway is in Basic mode.

- Issue ID 84986: If users log on with clientless access and attempt to open an external Web site (such as http://www.google.com) from the Email tab in the Access Interface, users might receive the Access Gateway logon page instead of the external Web site.

- Issue ID 88268: If users attempt to open a large Microsoft Word file from a Distributed File Share (DFS) hosted on Windows Server 2008 64-bit, the Access Gateway fails.

- Issue ID 81494: If users access a Distributed File Share on a computer running Windows Server 2008 64-bit, a blank folder appears in the directory path.

- Issue ID 83492: When users log on using clientless access, a JavaScript error might appear when the logon page opens.

- Issue ID 83819: If you configure a load balancing virtual server and the destination port is 21, when users log on with the Access Gateway Plug-in, logon is successful but data connections do not go through. When you configure a load balancing virtual server, do not use port 21.

- Issue ID 84894: When users log off from the Access Gateway Plug-in and then clear the cache in Internet Explorer and Firefox, users might receive an error message that says "Error. Not a privileged user." Access Gateway records an HTTP/1.1 403 Access Forbidden error message in the logs.

- Issue ID 84915: If users attempt to open and edit a Microsoft Office file from Outlook Web Access, users might receive an error and the file takes a long time to open. To allow users to edit files from Outlook Web Access, do the following:

  1. Create a clientless access Outlook Web Access Profile and enable persistent cookies.

  2. Bind the Outlook Web Outlook regular expression to this profile.

  3. Bind the profile so that is assumes the highest priority.
- Issue ID 86122: If you disable transparent interception and set the force time-out setting, when users log on with the Access Gateway Plug-in for Java, when the time-out period expires, a session time-out message appears on the user device, however the session is not terminated on Access Gateway.

- Issue ID 86123: If users log on with clientless access in the Firefox Web browser, when users click a link for a virtual application, the tab closes and the application does not

start. If users right-click the virtual application and attempt to open it in a new window, the Web Interface appears and users receive the warning "Published resource shortcuts are currently disabled." Users can open the virtual application in Internet Explorer.

- Issue ID 86323: If you configure single sign-on with Windows and configure the user name with special characters, when users log on to Windows 7 Professional, single sign-on fails. Users receive the error message "Invalid username or password. Please try again." This issue does not occur if users log on to Windows XP.
- Issue IDs 86470 and 86787: When users log on with the Access Gateway Plug-in for Windows using Internet Explorer 9, a delay may occur in establishing the connection. The Access Interface, or a custom home page, might take a long time to appear when users log on using Internet Explorer 9.
- Issue ID 86471: When users log on with the Access Gateway Plug-in by using a Web browser, users might see a delay during logon.
- Issue ID 86722: When users log on with clientless access using Internet Explorer 9 and connect to SharePoint 2007, some images might not appear correctly.

Application Firewall Issues

- Issue ID 83089: The users can look at the default signature rules in configuration utility, but it will be useful to have a comprehensive list of all the rules accessible in documentation or white papers for users to review.
- Issue ID 0259458: Attempts to upload a 30 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.

CloudBridge Issues

- Issue ID 91850 (nCore and nCore VPX): The NetScaler appliance drops TCP packets when the server has to send a full-size packet, that has the DF bit unset, across the cloud bridge. This happens because of bad checksum.

Command Line Interface Issues

- Issue ID 82908 (nCore): In certain rare cases, if the NetScaler MPX appliance is subject to conditions of heavy SSL-related traffic, CLI commands fail and report a configuration inconsistency error.

  Workaround: Check for configuration inconsistency by using the "show configstatus" command and reconfigure the appliance under low traffic conditions or during a maintenance period. If this does not resolve the issue, restart the appliance.

DataStream Issues

- Issue ID 83862 (nCore and nCore VPX): In this release, IPv6 addresses are not supported.

Domain Name System Issues

- Issue ID 93203 (nCore): A DNS policy that is bound to a GSLB service is not evaluated if the GSLB method is set to dynamic round trip time (RTT).

Load Balancing Issues

- Issue ID 82872: The setting of maximum requests per connection may be violated when a transaction is going on with the physical server.
- Issue ID 82929: When using a TCP monitor for a MYSQL service, the MySQL server blocks the MIP for making new connections.
- Issue ID 82996: The MYSQL monitor shows the service state as UP when no SNIP or MIP is configured.
- Issue ID 86096: While configuring the WI-EXTENDED monitor, the user will have to provide the value of sitepath in such a way that it does not end with a '/' . For example: add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb -domain ccc
- Issue ID 87201 (Classic and nCore): On a load balancing virtual server for TCP services on which stateful connection failover is enabled, an established connection may be broken if a failover occurs more than once while a large amount of data is being transferred.
- Issue ID 87407 (nCore and nCore VPX): When an RDP service is configured, the NetScaler appliance automatically maintains persistence through session cookies using Session Directory. You need not explicitly configure persistency on NetScaler. In the next releases, IP address based persistency will be supported. In some situations, (where multiple persons use same user-login credentials) session cookie persistence may not be helpful and IP-based persistence methods will be necessary. In some other situations, load balancing of the RDP services without persistence may be necessary. That is, each new connection to an RDP virtual server needs to be load balanced irrespective of a user's disconnected session existing on a terminal server.
- Issue ID 88593 (nCore): After failover, the 'maxclient' setting on a service is not honored.
- Issue ID 90271: The NetScaler appliance internally represents the servicegroup members with unique names. From the NetScaler release 9.3, the internal naming convention

changed because the delimiters used in the servicegroup member name are changed. The earlier format is: <service group name>_<IP address>_<port>. The new format is: <servicegroup name>?<IP address | server name>?<port>. Because of this change, application scripts that parse the servicegroup member name and extract the fields based on the delimiter "underscore" ("_"), will fail because the delimiter is now changed to "question mark" ("?").

NetScaler SDX Appliance Issues

- Issue ID 86597: The "Configuration" tab does not load on Internet Explorer version 9.0.

  Workaround: Run Internet Explorer version 9.0 in compatibility mode.

- Issue ID 88515: Client authentication is enabled by default in the Management Service VM. Consequently, HTTPS connections fail when you access the Management Service VM user interface from the Apple Safari browser.

- Issue ID 88556: While provisioning a NetScaler instance, if you have entered invalid NetScaler settings for any of the IP address, Netmask, or Gateway parameters, you cannot modify the values for these parameters later.

  Workaround: To correct the parameter values, log on to the NetScaler instance through the Xen Console. You also need to correct the values for this instance in the XenStore. After correcting the values in the both the places, rediscover the NetScaler instances from the Management Service VM user interface without selecting any specific instance, by clicking Rediscovery in the NetScaler Instance pane.

- Issue ID 89148: When you attempt to shut down the SDX appliance from the Management Service VM user interface, the appliance restarts instead of shutting down.

Platform Issues

- Issue ID 87419 (nCore): When you launch the remote console from the LOM configuration utility on the MPX 11500/13500/14500/16500/18500 appliance, remote keyboard redirection does not work.

  Workaround: Reset the LOM firmware. Note that the appliance may become unresponsive for approximately 60 seconds when you reset the LOM firmware. Warning: You should reset the LOM firmware only when one of the following conditions applies:

  - The appliance has just been installed.

  - The appliance is the secondary node in a high availability setup.
- Issue ID 90018 (nCore): When you upgrade any MPX appliance, except MPX 15000/17000, restart the appliance, and then apply the default configuration, the 1G interfaces are reset.

Reporting Issues

- Issue ID 85025 (nCore and nCore VPX): Reporting charts do not support plotting of counters per packet engine

SSL Issues

- Issue ID 80830 (nCore): If you attempt to delete an SSL certificate-key pair that is referenced by a certificate revocation list (CRL), the message, "ERROR: Configuration possibly inconsistent. Please check with the 'show configstatus' command or reboot," appears instead of the correct message. However, the correct message, "ERROR: Certificate is referenced by a CRL, OCSP responder, virtual server, service, or another certificate," appears upon subsequent attempts to delete the certificate-key pair.

- Issue ID 81850 (nCore): You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 10G FIPS appliance.

  Workaround: First, decrypt the key, and then import it. To decrypt the key, at the shell prompt, type: openssl rsa -in <EncryptedKey.key> > DecryptedKey.out

- Issue ID 85393: DSA certificate signed with the SHA-2 algorithm is not supported in the client authentication process.

- Issue ID 74279: The cipher TLS1-EXP1024-DES-CBC-SHA is not supported by the NetScaler appliance.

System Issues

- Issue ID 84099 (nCore): The NetScaler appliance may fail if traffic reaches a load balancing virtual server that uses the token method for load balancing and has connection failover enabled.

- Issue ID 84282: A global setting of less than 1220 for the maximum segment size (MSS) to use for TCP connections causes an excessive delay in saving the configuration.

- Issue ID 84320 (nCore and nCore VPX): The NetScaler appliance may fail if failover happens while high availability (HA) synchronization is in progress.

- Issue ID 94133: If a server (lb virtual server or cs vserver) is configured with the same IP address, port, and protocol as the server configured in the audit syslog or nslog action, the configured virtual server will be deleted on upgrading from 9.3_47.5 to 9.3_51.5.

  Workaround: Do the following:

  1. Remove the audit policy and action.

  2. Add the deleted virtual server.

3. Add the audit policy and action.

4. Save the configuration.

Web Interface Issues

- Issue ID 89052 (nCore and nCore VPX): The response from a Web Interface site, configured in direct mode, may have Java errors.

XML Issues

- Issue ID 81650: NetScaler import utility already does validation of XML Schema during import. But, it may fail to validate certain XHTML files while being imported as XMLSchema. These invalid XmlSchemas though, will be rejected if the user tries to use them in profile configuration (XMLValidation binding).

- Issue ID 82058: The 'unique' element in the XML schema is currently not supported.

- Issue ID 82059: The 'redefine' element in the XML schema is currently not supported.

- Issue ID 82069: When the Application Firewall validates XML messages, it does not validate the contents of elements that are defined as type "any" in the applicable XML schema. Specifically, it treats these elements as if the processContent attribute was set to "skip".

  Workaround: Replace the "any" type definitions in the XML schemas with definitions of the actual elements that occur in the XML message. (The "any" type is rarely used.)

- Issue ID 83707: Import feature for Schema and WSDL files does not support Non-ASCII characters. If an importing WSDL/Schema file contains Non-ASCII characters then it results in a partial import.

  Workaround: Convert XML schema or WSDL files to ASCII before importing them.

XML API Issues

- Issue ID 80170: The syntax of the "unset servicegroup" command has been changed to allow unsetting the parameters of the service group members. This can cause XML API incompatibility with respect to the "unset servicegroup" command.

Build 51.5

Release version: Citrix(R) NetScaler(R), version 9.3 build 51.5

Replaces build: None

Release date: August 2011

Language supported: English (US)

Readme version: 1.0

This readme is applicable for version 9.3 build 51.5 of the Citrix(R) NetScaler(R), Citrix NetScaler SDX, and Citrix Access Gateway(TM) software.

Review the following sections:

Note:

- Unless stated otherwise, this readme applies to all build types (Classic, nCore, and nCore VPX) of Citrix NetScaler and Citrix Access Gateway.
- A copy of this readme is also available in the Citrix Knowledge Center at http://support.citrix.com/.

Changes and Fixes

AAA Issues

- Issue ID 89587 (Classic and nCore): If you use Internet Explorer with the 'Display a notification for every script error' option enabled, when you access a virtual server on which AAA TM is configured, script error windows are displayed.

Access Gateway Issues

- Issue ID 85276: If Access Gateway does not receive responses from queries sent to servers running the Se Ticket Authority (STA), a new connection is opened for each STA query. When this occurs, Access Gatew fails.
- Issue ID 87967: If you configure LDAP authentication on Access Gateway with nested Group Extraction en when users log on with the Access Gateway Plug-in and the connection routes through a virtual IP addres authentication may fail due to a communication failure between Access Gateway and Active Directory.
- Issue ID 89116: When users log on with the Access Gateway Plug-in on computers running Windows XP an Vista, the Avaya IP Softphone application does not open.
- Issue ID 89641: If you configure group extraction and groups on the authorization server exceed 16 kiloby (KB), when users log on they might receive an HTTP 500 Internal server error message.
- Issue ID 89855: If you create a pre-authentication policy to check for a registry entry and use a large inte value, such as CLIENT.REG('HKEY_LOCAL_MACHINE\\\\SYSTEM\\\\CurrentControlSet\\\\Services\\\\Eventlog_RegCheck02') == 4052471216, user authentication fails.

AppFlow Issues

- Issue ID 91472 (nCore and nCore VPX): The AppFlow feature is re-enabled on a service when you disable it by using the respective Configure Service dialog box.
- Issue ID 92796 (nCore and nCore VPX): NetScaler appliance fails when attempting to export L7 AppFlow records to the collector because of an internal issue.

Application Firewall Issues

- Issue ID 81616: Attempts to upload a 10 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.
- Issue ID 89103: In some cases with Application Firewall enabled, malformed http requests can cause users to re-login.
- Issue ID 91607: Some internal counters were being incremented incorrectly. This has been fixed.
- Issue ID 91695: The issue is that AppFw does not check for XSS vulnerabilities in HTTP request headers if the attack pattern is percent encoded. Customer is seeing that XSS attack is not detected in the Cookie header in a request because the XSS attack is percent encoded.
- Issue ID 91738: When both AppFw and IC are enabled, for requests that match advanced profiles, if the responses are already cached in IC, then client connection may be reset intermittently.
- Issue ID 92143: Setting "-enableformtagging off" while using the "add appfw profile" command worked fine in 9.2 builds but started to trigger a false ERROR in 9.3 build onwards. As a result, If any add appfw profile command contained "enabledformtagging off", it failed due to this ERROR after upgrade resulting in loss of such profiles.
- Issue ID 92297: The issue is that AppFw runs out of memory when processing HTTP traffic and it is caused by a memory leak in AppFw HTML Form processing code.

CloudBridge Issues

- Issue ID 91805 (nCore and nCore VPX): NetScaler crashes while sending TCP RST to those devices in the cloud, whose connection it manages.

DataStream Issues

- Issue ID 89076 (nCore and nCore VPX): The "Database Users" subnode in the configuration utility that lets you configure your database user name and password on the NetScaler is now moved under the "System" node. Therefore, to add a database user by using the configuration utility, in the navigation pane, expand "System", and then click "Database Users".
- Issue ID 89643 (nCore and nCore VPX): If you have a service for SQL Server 2005 bound to a load balancing virtual server of type MS SQL, you cannot connect to the virtual server by using SQL Management Studio 2008.
- Issue ID 92022 (nCore and nCore VPX): The authentication response from MS SQL Server is not interpreted correctly, thus leading to connection failure.

- Issue ID 92075 (nCore and nCore VPX): The user name in MS SQL Server is not case sensitive, but NetScaler handles it as case sensitive, therefore, causing authentication failure.

High Availability Issues

- Issue ID 90067 (nCore and nCore VPX): If the name of an RTSP load balancing virtual server has more than 31 characters, the NetScaler appliance might fail.
- Issue ID 91790: While the configuration synchronization is in progress, if failover process is triggered and the current secondary appliance becomes the primary appliance, the 'clear config' propagates to the new secondary appliance.

Integrated Caching Issues

- Issue ID 90810: A cache miss occurs when a content-coding value (for example, "gzip" or "compress") in the Accept-Encoding header is accompanied by a quality value (or "qvalue"). Following is an example of an Accept-Encoding header that results in a cache miss:

  Accept-Encoding: gzip;q=1.0

Load Balancing Issues

- Issue ID 90211 (Classic, nCore, and nCore VPX): If USIP is enabled and an HTTP request with CONNECT method comes on an existing connection, connections to LB proxy servers get reset.
- Issue ID 90423: When using the WI EXTENDED MONITOR for monitoring the Web Interface services, in the response to the GET request, if the 'ASP' string is not sent in the first SET cookie, the monitor failed.
- Issue IDs 92081, 92140, and 92215 (nCore): In some rare cases, the NetScaler appliance fails.

NetScaler SDX Appliance Issues

- Issue ID 91335: In certain complex SDX configurations involving LA and HA, one or more 1G interfaces might not begin to receive traffic. A 'reset interface' is then needed to start RX traffic flowing. This is applicable to the e1kvf 1G interfaces only, not the management interfaces.

- Issue ID 91797: Two options, "Force Shut Down" and "Force Reboot" have been added that lets you shut down and restart a NetScaler instance forcefully. You can use these options if normal shut down and/or reboot operations are not working on a particular instance.

Networking Issues

- Issue ID 85290: The NetScaler appliance might fail if you remove an IP address from the appliance that is in the same subnet on which you have configured a SYSLOG or AUDITLOG server.
- Issue ID 85794: NetScaler appliance sends malformed BGP update messages, where Path attributes length value is not properly set, and overwrites withdrawn routes length. This happens because BGP uses circular buffers for sending messages and there was minor error during rewinding of buffer. Therefore, this problem is randomly observed during rewinding.
- Issue ID 91377: Port leak issue during passive FTP. This issue was observed when a client initiated a control connection, requested for a data connection but did not come back for data connection.

Platform Issues

- Issue ID 91829: Perl scripts using SSLeay do not function as expected if they are running on NetScaler release 9.3 build 50.3 and earlier.

Policies Issues

- Issue ID 91579: After the configuration is cleared on a standalone NetScaler appliance, the indexes in the built-in pattern set "ctx_file_extensions" are changed to incorrect values. Consequently, built-in cache policies like "ctx_images" and "ctx_web_css" are evaluated incorrectly. In a high availability setup, the issue also occurs in the secondary appliance after configuration synchronization and in both appliances after a failover.
- Issue ID 91660 (Classic): NetScaler Classic (non-nCore) systems might fail when evaluating the following policy-based entities:
  - An HTTP callout that includes a named expression.
  - A named expression that triggers an HTTP callout.
- Issue ID 92265: If an encrypted cookie value is truncated in an HTTP request, the NetScaler appliance may fail when attempting to decrypt the value. This applies to rewrite actions that use the ENCRYPT() and DECRYPT() functions and to Application Firewall cookie encryption.

Rewrite Issues

- Issue ID 87691: If rewrite is enabled, and if a server sends an amount of data that is more than the specified content length or includes data in the response body for responses that should not have a body (such as a 304 response), in some cases, rewrite does not work.

SSL Issues

- Issue ID 90331 (nCore): On the MPX 9700/10500/12500/15500 10G FIPS appliances in a high availability setup, key management commands such as "add ssl certkey" may fail while accessing the FIPS keys in the FIPS card. This may result in higher CPU utilization and a longer time for the secondary appliance to synchronize commands from the primary appliance.

System Issues

- Issue ID 92367: The value of allocated memory that is displayed for CONN_POOL in the output of the "nsconmsg -d memstats" command is incorrect. This issue is observed when a large amount of memory is allocated to CONN_POOL on NetScaler appliances that have large memory resources.

VPX Issues

- Issue ID 90689 (nCore VPX): The NetScaler VPX virtual appliance installed on the Citrix XenServer fails when it receives a frame with a packet size of more than 1514 bytes.
- Issue ID 92065 (nCore VPX): On a NetScaler VPX appliance, you cannot modify the HAmonitor or the tagall parameter for an interface by using the configuration utility.

Web Interface Issues

- Issue ID 85473 (nCore and nCore VPX): The show techsupport command is updated to collect the WebInterface.conf files from the NetScaler appliance.

Known Issues and Workarounds

Access Gateway Issues

- Issue ID 91832: If users logon with the Access Gateway Plug-in and then put the user device into hibernation, when the device resumes from a different network, the Access Gateway Plug-in reconnects, but when users log off, the default route might be deleted. Users can restart their device to obtain the network route.

- Issue ID 80175 and 82022: If you enable split tunneling, split DNS, and assign an intranet IP address on Access Gateway, when users log on with the Access Gateway Plug-in using a mobile broadband wireless device that uses the Sierra driver (for example, Telstra Compass or AT&T USBConnect) on a Windows 7 computer, Domain Name Service (DNS) resolution fails and the home page fails to open. You can use one of the following options to resolve the issue:

  1. Disable split tunneling

  2. Configure Access Gateway so user connections do not receive an intranet IP address.

  3. Configure the wireless device to use an Ethernet connection instead of a mobile broadband connection. For example:

  - Disable the setting Windows 7 Mobile Broadband in the Telstra Connection Manager Options dialog box.

  - Install Sierra Wireless Watcher (6.0.2849) if users connect with an AT&T USBConnect 881 network card. This installs an Ethernet adapter instead of the mobile broadband adapter.

  - Contact the manufacturer for other devices.
- Issue ID 89427: If users connect with the Access Gateway Plug-in by using an airtel 3G device and the Repeater Plug-in accelerates VPN traffic after the establishing the VPN connection, when users put the user device in standby or hibernate, when users resume the device, the Access Gateway Plug-in fails to reestablish the connection. When users disable acceleration with the Repeater Plug-in, restoration of the VPN connection is successful. Users can then enable acceleration with the Repeater Plug-in.

- Issue ID 89439: If users connect with the Access Gateway Plug-in and a T-Mobil 3G device and if you enable split tunneling and assign an intranet IP address to users on Access Gateway, users cannot connect to either intranet or external resources. To allow users to connect to both internal and external resources, disable split tunneling.

- Issue ID 89791: If users log on with a Windows-based computer that is not part of a domain by using a 3G network adapter and the Access Gateway Plug-in for Windows, requests that use the host name fail. In this instance, use the fully qualified domain name (FQDN) instead of the host name.

- Issue ID 90675: If users log on with the Access Gateway Plug-in for Windows and then access a CIFS share by using the Run dialog box, when users navigate to a folder in the share and attempt to copy a file to another file share, users receive an error message and the attempt fails.

- Issue ID 84787: When you issue the command "sh vpn vserver" on Access Gateway, the number of current ICA connections does not appear when Access Gateway is in Basic mode.

- Issue ID 84986: If users log on with clientless access and attempt to open an external Web site (such as http://www.google.com) from the Email tab in the Access Interface, users might receive the Access Gateway logon page instead of the external Web site.

- Issue ID 88268: If users attempt to open a large Microsoft Word file from a Distributed File Share (DFS) hosted on Windows Server 2008 64-bit, the Access Gateway fails.

- Issue ID 81494: If users access a Distributed File Share on a computer running Windows Server 2008 64-bit, a blank folder appears in the directory path.

- Issue ID 83492: When users log on using clientless access, a JavaScript error might appear when the logon page opens.

- Issue ID 83819: If you configure a load balancing virtual server and the destination port is 21, when users log on with the Access Gateway Plug-in, logon is successful but data connections do not go through. When you configure a load balancing virtual server, do not use port 21.

- Issue ID 84894: When users log off from the Access Gateway Plug-in and then clear the cache in Internet Explorer and Firefox, users might receive an error message that says "Error. Not a privileged user." Access Gateway records an HTTP/1.1 403 Access Forbidden error message in the logs.

- Issue ID 84915: If users attempt to open and edit a Microsoft Office file from Outlook Web Access, users might receive an error and the file takes a long time to open. To allow users to edit files from Outlook Web Access, do the following:

  1. Create a clientless access Outlook Web Access Profile and enable persistent cookies.

  2. Bind the Outlook Web Outlook regular expression to this profile.

  3. Bind the profile so that is assumes the highest priority.

- Issue ID 85861: If you enable ICA Proxy on Access Gateway, when users log on and attempt to open a virtual application, the connection to the Web Interface through Access Gateway times out and closes.

- Issue ID 85906: When users log on with an earlier version of the Access Gateway Plug-in, users do not receive the upgrade prompt and the user device receives a session ID. However, the session is not established and the Web browser trying to load the file services.html and upgrading the plug-in both fail.

- Issue ID 86122: If you disable transparent interception and set the force time-out setting, when users log on with the Access Gateway Plug-in for Java, when the time-out period expires, a session time-out message appears on the user device, however the session is not terminated on Access Gateway.

- Issue ID 86123: If users log on with clientless access in the Firefox Web browser, when users click a link for a virtual application, the tab closes and the application does not start. If users right-click the virtual application and attempt to open it in a new window, the Web Interface appears and users receive the warning "Published resource shortcuts are currently disabled." Users can open the virtual application in Internet Explorer.

- Issue ID 86323: If you configure single sign-on with Windows and configure the user name with special characters, when users log on to Windows 7 Professional, single sign-on fails. Users receive the error message "Invalid username or password. Please try again." This issue does not occur if users log on to Windows XP.

- Issue ID 86470 and 86787: When users log on with the Access Gateway Plug-in for Windows using Internet Explorer 9, a delay may occur in establishing the connection. The Access Interface, or a custom home page, might take a long time to appear when users log on using Internet Explorer 9.

- Issue ID 86471: When users log on with the Access Gateway Plug-in by using a Web browser, users might see a delay during logon.

- Issue ID 86722: When users log on with clientless access using Internet Explorer 9 and connect to SharePoint 2007, some images might not appear correctly.

Application Firewall Issues

- Issue ID 83089: The users can look at the default signature rules in configuration utility, but it will be useful to have a comprehensive list of all the rules accessible in documentation or white papers for users to review.

- Issue ID 86782: Importing appfw objects such as signatures, htmlerror page, and so on, may fail on first attempt if the import command contains a very long source URL. The command will succeed if executed a second time.

- Issue ID 0259458: Attempts to upload a 30 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.

CloudBridge Issues

- Issue ID 91850 (nCore and nCore VPX): NetScaler drops TCP packets when the server has to send a full-size packet, that has the DF bit unset, across the cloud bridge. This happens due to bad checksum.

Command Line Interface Issues

- Issue ID 82908 (nCore and nCore VPX): In certain rare cases, when the NetScaler appliance is subject to conditions of heavy SSL-related traffic, CLI commands fail and report a configuration inconsistency error.

  Workaround: Check for configuration inconsistency by using the "show configstatus" command and reconfigure the appliance under low traffic conditions or during a maintenance period. If this does not resolve the issue, restart the appliance.

DataStream Issues

- Issue ID 83862 (nCore and nCore VPX): In this release, IPv6 addresses are not supported.

Load Balancing Issues

- Issue ID 82872: The setting of maximum requests per connection may be violated when a transaction is going on with the physical server.
- Issue ID 82929: When using a TCP monitor for a MYSQL service, the MySQL server blocks the MIP for making new connections.
- Issue ID 82996: The MYSQL monitor shows the service state as UP when no SNIP or MIP is configured.
- Issue ID 86096: While configuring the WI-EXTENDED monitor, the user will have to provide the value of sitepath in such a way that it does not end with a '/' .

  For example:

  add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa

  -password bbb -domain ccc
- Issue ID 87201 (Classic and nCore): On a load balancing virtual server for TCP services on which stateful connection failover is enabled, an established connection may be broken if a failover occurs more than once while a large amount of data is being transferred.
- Issue ID 87407 (nCore and nCore VPX): When an RDP service is configured, the NetScaler appliance automatically maintains persistence through session cookies using Session Directory. You need not explicitly configure persistency on NetScaler. In the next releases, IP address based persistency will be supported.

  In some situations, (where multiple persons use same user-login credentials) session cookie persistence may not be helpful and IP-based persistence methods will be necessary. In some other situations, load balancing of the RDP services without persistence may be necessary. That is, each new connection to an RDP virtual server needs to be load balanced irrespective of a user's disconnected session existing on a terminal server.

- Issue ID 88593 (nCore): After failover, the 'maxclient' setting on a service is not honored.

NetScaler SDX Appliance Issues

- Issue ID 88515: Client authentication is enabled by default in the Management Service VM. Consequently, HTTPS connections fail when you access the Management Service VM user interface from the Apple Safari browser.
- Issue ID 88556: While provisioning a NetScaler instance, if you have entered invalid NetScaler settings for the IP address, Netmask, or Gateway parameters, you cannot modify the values for these parameters later.

  Workaround: To rectify the parameter values, log on to the NetScaler instance through the Xen Console. You also need to rectify the values for this instance in the XenStore.

  After correcting in the both the places, rediscover the NetScaler instances from the Management Service VM user interface without selecting any specific instance by clicking Rediscovery in the NetScaler Instance pane.
- Issue ID 89148: When you attempt to shut down the SDX appliance from the Management Service VM user interface, the appliance restarts instead of shutting down.

Platform Issues Issues

- Issue ID 87419 (nCore): When you launch the remote console from the LOM configuration utility on the MPX 11500/13500/14500/16500/18500 appliance, remote keyboard redirection does not work.

  Workaround: Reset the LOM firmware. Note that the appliance may become unresponsive for approximately 60 seconds when you reset the LOM firmware.

  Warning: You should reset the LOM firmware only when one of the following conditions apply:

  - The appliance has just been installed.
  - The appliance is the secondary node in a high availability setup.
- Issue ID 90018 (nCore): When you upgrade any MPX appliance, except MPX 15000/17000, restart the appliance, and then apply the default configuration, the 1G interfaces are reset.

Reporting Issues

- Issue ID 85025 (nCore and nCore VPX): Reporting charts do not support plotting of counters per packet engine

SSL Issues

- Issue ID 80830 (nCore): When you attempt to delete an SSL certificate key-pair object that is referenced by a Certificate Revocation List (CRL), the message, "ERROR: Configuration possibly inconsistent. Please check with the "show configstatus" command or reboot," is displayed.

  This message is not the intended message. However, on the subsequent attempt to delete the certificate key-pair object, the correct message, "ERROR: Certificate is referenced by a CRL, OCSP responder, virtual server, service, or another certificate," is displayed.

- Issue ID 81850 (nCore): You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 10G FIPS appliance.

  Workaround: First, decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

  openssl rsa -in <EncryptedKey.key> > DecryptedKey.out

- Issue ID 85393: DSA certificate signed with SHA-2 algorithm is not supported in the client authentication process.

- Issue ID 74279: The cipher TLS1-EXP1024-DES-CBC-SHA is not supported by the NetScaler appliance.

System Issues

- Issue ID 84099 (nCore): The NetScaler appliance may fail if traffic reaches a load balancing virtual server on which connection failover is enabled and the load balancing method is the token method.

- Issue ID 84282: If the global setting for the maximum segment size (MSS) to use for TCP connections is less than 1220, the NetScaler appliance causes an excessive delay in saving the configuration.

- Issue ID 84320: The NetScaler appliance may fail if failover happens while high availability (HA) synchronization is in progress.

Web Interface Issues

- Issue ID 89052 (nCore and nCore VPX): The response from a Web Interface site, configured in direct mode, may have Java errors.

XML Issues

- Issue ID 81650: NetScaler import utility already does validation of XML Schema during import. But, it may fail to validate certain XHTML files while being imported as XMLSchema. These invalid XmlSchemas though, will be rejected if the user tries to use them in profile configuration (XMLValidation binding).
- Issue ID 82058: The 'unique' element in the XML schema is currently not supported.
- Issue ID 82059: The 'redefine' element in the XML schema is currently not supported.
- Issue ID 82069: When the Application Firewall validates XML messages, it does not validate the contents of elements that are defined as type "any" in the applicable XML schema. Specifically, it treats these elements as if the processContent attribute was set to "skip".

  Workaround: Replace the "any" type definitions in the XML schemas with definitions of the actual elements that occur in the XML message. (The "any" type is rarely used.)
- Issue ID 83707: Import feature for Schema and WSDL files does not support Non-ASCII characters. If an importing WSDL/Schema file contains Non-ASCII characters then it results in a partial import.Workaround: Convert XML schema or WSDL files to ASCII before importing them.

XML API Issues

- Issue ID 80170: The syntax of the unset servicegroup command has been changed to allow unsetting of the parameters of the service group members. This can cause XML API incompatibility with respect to the unset servicegroup command.

Build 50.3

Release version: Citrix(R) NetScaler(R), version 9.3 build 50.3

Replaces build: None

Date: July 2011

Language supported: English (US)

Readme version: 1.0

This section captures all the changes, bug fixes, and known issues in Citrix NetScaler release 9.3 build 50.3.

Review the following sections:

- Changes and Fixes
- Known Issues and Workarounds

Note:

- Unless stated otherwise, all changes, bug fixes, and known issues in this readme apply to Citrix(R) NetScaler(R) 9.3 Classic, NetScaler 9.3 nCore(TM), and NetScaler VPX(TM) 9.3. For Access Gateway, unless stated otherwise, the known issues, changes, and fixed issues apply to 9.3 Classic and nCore builds.
- The latest version of the readme is available in the Citrix Knowledge Center at http://support.citrix.com/.

Changes and Fixes

Access Gateway Issues

- Issue ID 81781: When users connect to Access Gateway and change an expired password on the challenge response page, users can enter up to 256 characters. If users create a password with more than 31 characters, however, when they log on again, Access Gateway displays a 401 authentication error and logon fails.
- Issue ID 87050: When users log on with Citrix online plug-ins and you configure Access Gateway in a high availability pair, if the primary appliance fails, occasionally the connection fails and users receive an error message stating that the connection to the appliance is interrupted.
- Issue ID 90098: If there is a large amount of network traffic through the Access Gateway VPN tunnel and if users access 40,000 or more resources through the tunnel, access to new resources fail.

Application Firewall Issues

- Issue ID 89103: In some cases with Application Firewall enabled malformed http requests can cause users to re-login.
- Issue ID 89417: Relaxation rules for transformation of SQL special characters do not work in some cases.
- Issue ID 90909: Config sync is triggered every minute instead of exponentially backing off when files are missing in secondary required by the configuration.

Cache Redirection Issues

- Issue ID 91008: If the cache redirection virtual server receives a request without a backslash (/) after the hostname (at the end of a request), the request gets corrupted when it is sent to the destination. When the NetScaler sends the request to the destination, the space is missed between '/' and 'HTTP/1.1'.
- Issue ID 91062: If forward proxy is configured, NetScaler always connects to the physical server using port 443 instead of using the port specified in the client request, and there is a failure in serving the request.

CloudBridge Issues

- Issue ID 89428 (nCore and nCore VPX): This build supports the NAT implementation of RFC 3947 and 3948 for the cloud bridge peers to communicate properly when any of the peer is behind a NAT device. For more information about configuring a cloud bridge, see the "Cloud Bridge" chapter of the Citrix NetScaler Networking Guide at http://support.citrix.com/article/CTX128671.

Configuration Utility Issues

- Issue ID 88379: Now "NetScaler Management Pack for System Center Operation Manager 2007 (SCOM)" can be downloaded from NetScaler GUI Downloads page. The Citrix NetScaler Operation Manager pack provides monitors and rules to monitor the NetScaler systems deployed in your network. The Citrix NetScaler Performance and Resource Optimization (PRO) Management Pack (MP) provides monitors and rules to monitor the health of the virtual servers configured on the managed NetScaler systems and initiate corrective actions using the PRO feature of SCVMM when the virtual servers become unhealthy.

Content Switching Issues

- Issue ID 89906: The NetScaler appliance does not support content switching based on the parameters of a stored procedure call for database protocols.

DataStream Issues

- Issue ID 88227 (nCore): The NetScaler appliance does not support the MySQL server version 5.5.7 or later. The NetScaler fails when you log on to the MySQL load balancing virtual server.
- Issue ID 90228 (nCore and nCore VPX): You can use the following expressions to configure content switching based on remote procedure call (RPC) names or IDs:
  - MSSQL.REQ.RPC.NAME. Returns the name of the procedure that is being called in a remote procedure call (RPC) request. The name is returned as a string.
  - MSSQL.REQ.RPC.IS_PROCID. Returns a Boolean value that indicates whether the remote procedure call (RPC) request contains a process ID or an RPC name. A return value of TRUE indicates that the request contains a process ID and a return value of FALSE indicates that the request contains an RPC name.
  - MSSQL.REQ.RPC.PROCID. Returns the process ID of the remote procedure call (RPC) request as an integer.

- Issue ID 91217: For MySQL virtual servers, the NetScaler appliance does not correctly handle a query of the format 'SET NAMES 'UTF8" with the character set name in quotation marks. This causes the further requests on the same connection to fail.

EdgeSight Monitoring Issues

- Issue ID 90241: When EdgeSight monitoring is enabled on a LB/CS VIP, the user is given options to choose if the responses to the clients from that VIP should be compressed or not. The decision to bind/unbind compression policies to that VIP will be taken accordingly.

Integrated Caching Issues

- Issue ID 89374 (Classic): If all the three following conditions are met, when the NetScaler appliance receives a request for an object and it attempts to re-validate and serve, NetScaler fails:
  - The content group setting "alwaysEvalPolicies" is set to YES.
  - The response cached in this group has status codes greater than 300.
  - The object is in expired state.

Load Balancing Issues

- Issue ID 81582: When an SNIP address is configured as an ADNS service and later the ADNS servers IP address is changed, the ADNS count for the old IP address does not get decremented, and an error occurs when you try to remove the old IP address. Example:

  add ip 1.1.1.1 255.255.255.0 [configured as SNIP]

  add service adns 1.1.1.1 adnS 53 [configuring the same IP as ADNS]

  set server 1.1.1.1 -ipaddress 1.1.1.2 [changing the adns server IP to new IP]

  rm ip 1.1.1.1 [Removing old IP returns error]

- Issue ID 82185 (nCore): In low-traffic-rate scenarios as mentioned below, the least connection load balancing becomes uneven. For example, when a service gets very low traffic rate such as two or three requests per second per service, and the service takes an average of 800 milliseconds to respond with the first byte. To make the load balancing precise and even in low-traffic scenarios, the following option is provided:

  set lb parameter -consolidatedLConn (YES | NO)

By default, the option is set to 'Yes'. If and only if there is uneven least connection load balancing in low-traffic scenarios, you can set the "consolidatedLConn" to 'No' to make the load balancing even.

- Issue ID 82571: When a load balancing virtual server and content switching virtual server are configured on the NetScaler appliance, when a server connection terminates, some counters on the load balancing virtual server such as open established connections (OEs) are not decremented until the connection is flushed. This may lead to other side effects like unnecessary spillover.
- Issue ID 87893: If you set the maxclient value very low, the NetScaler appliance closes connections frequently in spite of reusing them.
- Issue ID 89697 (nCore): If all the Branch Repeater appliances bound to a load balancing virtual server are DOWN, the NetScaler appliance should bypass the Branch Repeater appliances and send traffic directly to the data center.
- Issue ID 90426 (nCore and nCore VPX): The MS-SQL ECV monitor may have errors when the expected response is a result set.
- Issue ID 90917: When you create a load balancing monitor of type MSSQL-ECV, the default expression prefix is now changed from HTTP to MSSQL. When you create a load balancing monitor of type MySQL-ECV, the default expression prefix is now changed from HTTP to MySQL.

NetScaler SDX Appliance Issues

- Issue ID 90966: While modifying nsroot user if password contains special characters such as $, then password is not correctly updated on hypervisor and management vm can not communicate with hypervisor anymore.

Networking Issues

- Issue ID 88583: When OSPF authentication is in use and the packet size is 512, the authentication digest verification on the NetScaler can go wrong resulting in dropped packets.

NITRO API Issues

- Issue ID 90781: In getlbvserver, cookieipport information is missing for servicegroup bindings. In lbvserver response structure, servicegroup member information is missing.

Platform Issues

- Issue ID 88559 (nCore): On the MPX 17500/19500/21500 and MPX 11500/13500/14500/16500/18500 appliances, programming the IP address, default gateway, and netmask from the front panel keypad does not work. Note: You can use the keypad for this purpose only when the appliance has a factory default configuration.
- Issue ID 91248 (nCore): The following table shows the maximum throughput available on the Citrix NetScaler MPX 11500/13500/14500/16500/18500 appliances.

  Model Maximum throughput (in Gbps)

  11500 8

  13500 12

  14500 18

  16500 24

  18500 36

SNMP Issues

- Issue ID 90440: An snmp request from a manager will not get a response from the NetScaler if an rnat rule has been configured on the NetScaler for the manager's subnet with a SNIP as natip and that SNIP has dynamic routing enabled on it.

SSL Issues

- Issue ID 89491: If a policy for client authentication during renegotiation over SSLv3 protocol is configured on the backend server, the NetScaler fails during SSL renegotiation.

System Issues

- Issue ID 88885 (nCore and nCore VPX): During race conditions between user logins and session timeouts on the NetScaler appliance, if a core-to-core message for logout request handling fails, the core that receives the logout message might not clean up the

user session. When a user whose session has not been cleaned up logs on to the appliance again, session duplication occurs on the core and the appliance might fail.

- Issue ID 89527: The NetScaler appliance fails when a large number of HTTP pipeline POST requests with large content lengths are received over the same client-side connection.

- Issue ID 89864: When a server does not receive a window update sent by the NetScaler appliance, download latency is observed.

- Issue ID 89986 (nCore and nCore VPX): If addition of an IP on NetScaler was failing in one of the PE's and succeeding in other PE's, it would lead to config inconsistency across PE's in NetScaler. Now we have added proper recovery mechanism to recover from this failure where if "add ip" command fails on one of PE, we will revert this command across all the successful PE's also.

- Issue ID 90715 (nCore): If the channel is Down/Disable, sh channel command always gives channel downtime as 0h00m00s i.e downtime not increasing. However sh interface gives the correct channel downtime.

Web Interface Issues

- Issue ID 90121 (nCore and nCore VPX): Launching of XenApp application fails on iPad when using Web Interface on NetScaler through Safari with error "Unable to download file". The root cause of this issue is that Safari on iPad does not pass the downloaded ica file to Citrix Receiver correctly since file extension is jsp. In 9.3 build 50.1 onwards, this issue has been fixed by configuring rewrite policy in WI wizard which changes the file extension to .ica while downloading the ICA file.

- Issue ID 90658 (nCore and nCore VPX): If vpn vserver is configured on port other than 443, Single Sign On from Access Gateway to Web Interface fails and Web Interface login remains stuck with blank page at agesso.jsp when logging in to the Web Interface through Access Gateway. Root cause of this issue was incorrect port configuration in AGEWebServiceURL within WebInterface.conf for Web Interface site. Also, DNS record for Access Gateway VIP was not added correctly. This issue is resolved in 9.3 build 50.x onwards.

XML Issues

- Issue ID 68633: You cannot set the total import size limit to less than the currently imported object size.

Known Issues and Workarounds

Access Gateway Issues

- Issue ID 80175 and 82022: If you enable split tunneling, split DNS, and assign an intranet IP address on Access Gateway, when users log on with the Access Gateway Plug-in using a mobile broadband wireless device that uses the Sierra driver (for example, Telstra Compass or AT&T USBConnect) on a Windows 7 computer, Domain Name Service (DNS) resolution fails and the home page fails to open. You can use one of the following options to resolve the issue:

    1. Disable split tunneling.

    2. Configure Access Gateway so user connections do not receive an intranet IP address.

    3. Configure the wireless device to use an Ethernet connection instead of a mobile broadband connection. For example:

        - Disable the setting Windows 7 Mobile Broadband in the Telstra Connection Manager Options dialog box.

        - Install Sierra Wireless Watcher (6.0.2849) if users connect with an AT&T USBConnect 881 network card. This installs an Ethernet adapter instead of the mobile broadband adapter.

        - Contact the manufacturer for other devices.
- Issue ID 89427: If users connect with the Access Gateway Plug-in by using an airtel 3G device and the Repeater Plug-in accelerates VPN traffic after the establishing the VPN connection, when users put the user device in standby or hibernate, when users resume the device, the Access Gateway Plug-in fails to reestablish the connection. When users disable acceleration with the Repeater Plug-in, restoration of the VPN connection is successful. Users can then enable acceleration with the Repeater Plug-in.

- Issue ID 89439: If users connect with the Access Gateway Plug-in and a T-Mobil 3G device and if you enable split tunneling and assign an intranet IP address to users on Access Gateway, users cannot connect to either intranet or external resources. To allow users to connect to both internal and external resources, disable split tunneling.

- Issue ID 89791: If users log on with a Windows-based computer that is not part of a domain by using a 3G network adapter and the Access Gateway Plug-in for Windows, requests that use the host name fail. In this instance, use the fully qualified domain name (FQDN) instead of the host name.

- Issue ID 90675: If users log on with the Access Gateway Plug-in for Windows and then access a CIFS share by using the Run dialog box, when users navigate to a folder in the share and attempt to copy a file to another file share, users receive an error message and the attempt fails.

- Issue ID 84787: When you issue the command "sh vpn vserver" on Access Gateway, the number of current ICA connections does not appear when Access Gateway is in Basic mode.

- Issue ID 84986: If users log on with clientless access and attempt to open an external Web site (such as http://www.google.com) from the Email tab in the Access Interface, users might receive the Access Gateway logon page instead of the external Web site.

- Issue ID 88268: If users attempt to open a large Microsoft Word file from a Distributed File Share (DFS) hosted on Windows Server 2008 64-bit, the Access Gateway fails.

- Issue ID 81494: If users access a Distributed File Share on a computer running Windows Server 2008 64-bit, a blank folder appears in the directory path.

- Issue ID 83492: When users log on using clientless access, a JavaScript error might appear when the logon page opens.

- Issue ID 83819: If you configure a load balancing virtual server and the destination port is 21, when users log on with the Access Gateway Plug-in, logon is successful but data connections do not go through. When you configure a load balancing virtual server, do not use port 21.

- Issue ID 84894: When users log off from the Access Gateway Plug-in and then clear the cache in Internet Explorer and Firefox, users might receive an error message that says "Error. Not a privileged user." Access Gateway records an HTTP/1.1 403 Access Forbidden error message in the logs.

- Issue ID 84915: If users attempt to open and edit a Microsoft Office file from Outlook Web Access, users might receive an error and the file takes a long time to open. To allow users to edit files from Outlook Web Access, do the following:

    1. Create a clientless access Outlook Web Access Profile and enable persistent cookies.

    2. Bind the Outlook Web Outlook regular expression to this profile.

    3. Bind the profile so that is assumes the highest priority.
- Issue ID 85861: If you enable ICA Proxy on Access Gateway, when users log on and attempt to open a virtual application, the connection to the Web Interface through Access Gateway times out and closes.

- Issue ID 85906: When users log on with an earlier version of the Access Gateway Plug-in, users do not receive the upgrade prompt and the user device receives a session ID. However, the session is not established and the Web browser trying to load the file services.html and upgrading the plug-in both fail.

- Issue ID 86022: If you configure the user device to enable users to log on only using the Access Gateway Plug-in and then change the plug-in Web address to an unresolvable address, when users try to log on through the logon dialog box, an authentication error appears. Then, if users try to log on using the plug-in, the logon dialog box does not appear and users cannot change the Web address. Users should exit and then restart the plug-in to subsequently change the Web address.

- Issue ID 86122: If you disable transparent interception and set the force time-out setting, when users log on with the Access Gateway Plug-in for Java, when the time-out period expires, a session time-out message appears on the user device, however the

session is not terminated on Access Gateway.

- Issue ID 86123: If users log on with clientless access in the Firefox Web browser, when users click a link for a virtual application, the tab closes and the application does not start. If users right-click the virtual application and attempt to open it in a new window, the Web Interface appears and users receive the warning "Published resource shortcuts are currently disabled." Users can open the virtual application in Internet Explorer.

- Issue ID 86323: If you configure single sign-on with Windows and configure the user name with special characters, when users log on to Windows 7 Professional, single sign-on fails. Users receive the error message "Invalid username or password. Please try again." This issue does not occur if users log on to Windows XP.

- Issue ID 86470 and 86787: When users log on with the Access Gateway Plug-in for Windows using Internet Explorer 9, a delay may occur in establishing the connection. The Access Interface, or a custom home page, might take a long time to appear when users log on using Internet Explorer 9.

- Issue ID 86471: When users log on with the Access Gateway Plug-in by using a Web browser, users might see a delay during logon.

- Issue ID 86722: When users log on with clientless access using Internet Explorer 9 and connect to SharePoint 2007, some images might not appear correctly.

Application Firewall Issues

- Issue ID 81616: Attempts to upload a 10 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.

- Issue ID 83089: The users can look at the default signature rules in configuration utility, but it will be useful to have a comprehensive list of all the rules accessible in documentation or white papers for users to review.

- Issue ID 86782: Importing appfw objects such as signatures, htmlerror page, and so on, may fail on first attempt if the import command contains a very long source URL. The command will succeed if executed a second time.

CloudBridge Issues

- Issue ID 91805 (nCore and nCore VPX): NetScaler crashes while sending TCP RST to those devices in the cloud, whose connection it manages.

- Issue ID 91850 (nCore and nCore VPX): NetScaler drops TCP packets when the server has to send a full-size packet, that has the DF bit unset, across the cloud bridge. This happens due to bad checksum.

Command Line Interface Issues

- Issue ID 82908: In certain rare cases, when the NetScaler appliance is subject to conditions of heavy SSL-related traffic, CLI commands fail and report a configuration inconsistency error.

  Workaround: Check for configuration inconsistency by using the "show configstatus" command and reconfigure the appliance under low traffic conditions or during a maintenance period. If this does not resolve the issue, restart the appliance.

DataStream Issues

- Issue ID 83862 (nCore, nCore VPX): In this release, IPv6 addresses are not supported.

Integrated Caching Issues

- Issue ID 81159: When the NetScaler appliance receives a single byte-range request, if the starting position of the range is beyond 9 megabytes, the appliance sends the client a full response with a status code of 200 OK instead of a partial response.

Load Balancing Issues

- Issue ID 82872: The setting of maximum requests per connection may be violated when a transaction is going on with the physical server.
- Issue ID 82929: When using a TCP monitor for MYSQL service, the MySQL server blocks the MIP for making new connections.
- Issue ID 82996: The MYSQL monitor shows the service state as UP when no SNIP or MIP is configured.
- Issue ID 86096: While configuring the WI-EXTENDED monitor, the user will have to provide the value of sitepath in such a way that it does not end with a '/' .

  For example:

  add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb -domain ccc
- Issue ID 87201 (Classic, nCore): On a load balancing virtual server for TCP services on which stateful connection failover is enabled, an established connection may be broken if a failover occurs more than once while a large amount of data is being transferred.

- Issue ID 87407 (nCore and nCore VPX): When an RDP service is configured, the NetScaler appliance automatically maintains persistence through session cookies using Session Directory. You need not explicitly configure persistency on NetScaler. In the next releases, IP address based persistency will be supported. In some situations, (where multiple persons use same user-login credentials) session cookie persistence may not be helpful and IP-based persistence methods will be necessary. In some other situations, load balancing of the RDP services without persistence may be necessary. That is, each new connection to an RDP virtual server needs to be load balanced irrespective of a user's disconnected session existing on a terminal server.

NetScaler SDX Appliance Issues

- Issue ID 88515: Client authentication is enabled by default in the Management Service VM. Consequently, HTTPS connections fail when you access the Management Service VM user interface from the Apple Safari browser.

- Issue ID 88556: While provisioning a NetScaler instance, if you have entered invalid NetScaler settings for the IP address, Netmask, or Gateway parameters, you cannot modify the values for these parameters later.

  Workaround: To rectify the parameter values, log on to the NetScaler instance through the Xen Console. You also need to rectify the values for this instance in the XenStore. After correcting in the both the places, rediscover the NetScaler instances from the Management Service VM user interface without selecting any specific instance by clicking Rediscovery in the NetScaler Instance pane.

- Issue ID 89148: When you attempt to shut down the SDX appliance from the Management Service VM user interface, the appliance restarts instead of shutting down.

- Issue ID 91335: In certain complex SDX configurations involving LA and HA, one or more 1G interfaces might not begin to receive traffic. A "reset interface" is then needed to start RX traffic flowing. This is applicable to the e1kvf 1G interfaces only, not the management interfaces.

Platform Issues

- Issue ID 87419 (nCore): When you launch the remote console from the LOM configuration utility on the MPX 11500/13500/14500/16500/18500 appliance, remote keyboard redirection does not work.

  Workaround: Reset the LOM firmware. Note that the appliance may become unresponsive for approximately 60 seconds when you reset the LOM firmware. Warning: You should reset the LOM firmware only when one of the following conditions apply:

  - The appliance has just been installed.

  - The appliance is the secondary node in a high availability setup.

- Issue ID 90018 (nCore): When you upgrade any MPX appliance, except MPX 15000/17000, to release 9.3 build 50.3, restart the appliance, and then apply the default configuration, the 1G interfaces are reset.

Reporting Issues

- Issue ID 85025 (nCore, nCore VPX): Reporting charts do not support plotting of counters per packet engine.

SSL Issues

- Issue ID 74279: The cipher TLS1-EXP1024-DES-CBC-SHA is not supported by the NetScaler appliance.
- Issue ID 80830 (nCore): When you attempt to delete an SSL certificate key-pair object that is referenced by a Certificate Revocation List (CRL), the message, "ERROR: Configuration possibly inconsistent. Please check with the "show configstatus" command or reboot," is displayed. This message is not the intended message. However, on the subsequent attempt to delete the certificate key-pair object, the correct message, "ERROR: Certificate is referenced by a CRL, OCSP responder, virtual server, service, or another certificate," is displayed.
- Issue ID 81850 (nCore): You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/ 12500/15500 10G FIPS appliance.

  Workaround: First, decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

  openssl rsa -in <EncryptedKey.key> > DecryptedKey.out

System Issues

- Issue ID 84099 (nCore): The NetScaler appliance may fail if traffic reaches a load balancing virtual server on which connection failover is enabled and the load balancing method is the token method.
- Issue ID 84282: If the global setting for the maximum segment size (mss) to use for TCP connections is less than 1220, the NetScaler appliance causes excessive delay to save the configuration.
- Issue ID 84320 (nCore): The NetScaler appliance may fail if failover happens while high availability (HA) synchronization is in progress.
- Issue ID 88593 (nCore): After failover, the maxclient configuration on a service is not honored.

Web Interface Issues

- Issue ID 89052 (nCore and nCore VPX): The response from a Web Interface site, configured in direct mode, may have Java errors.

XML Issues

- Issue ID 81650: NetScaler import utility already does validation of XML Schema during import. But, it may fail to validate certain XHTML files while being imported as XMLSchema. These invalid XmlSchemas though, will be rejected if the user tries to use them in profile configuration (XMLValidation binding).
- Issue ID 82058: The 'unique' element in the XML schema is currently not supported.
- Issue ID 82059: The 'redefine' element in the XML schema is currently not supported.
- Issue ID 82069: When the Application Firewall validates XML messages, it does not validate the contents of elements that are defined as type "any" in the applicable XML schema. Specifically, it treats these elements as if the processContent attribute was set to "skip".

  Workaround: Replace the "any" type definitions in the XML schemas with definitions of the actual elements that occur in the XML message. (The "any" type is rarely used.)
- Issue ID 83707: Import feature for Schema and WSDL files does not support Non-ASCII characters. If an importing WSDL/Schema file contains Non-ASCII characters then it results in a partial import.

  Workaround: Convert XML schema or WSDL files to ASCII before importing them.

XML API Issues

- Issue ID 80170: The syntax of the unset servicegroup command has been changed to allow unsetting of the parameters of the service group members. This can cause XML API incompatibility with respect to the unset servicegroup command.

Enhancement Releases

This section provides the readmes for enhancement releases of NetScaler version 9.3.

- Build 53.5006.e
- Build 51.5006.e
- Build 50.3002.e

Build 53.5006.e

Release version: Citrix(R) NetScaler(R), version 9.3.e, Build 53.5006.e

Replaces build: None

Readme version: 1.0

Release date: December 2011

Language supported: English (US)

This section describes the enhancements and known issues in version 9.3.e build 53.5006.e of the Citrix(R) NetScaler(R) software. For the list of bug fixes in this release, see the readme for version 9.3 build 53.5, which is available in the Citrix eDocs at http://support.citrix.com/proddocs/topic/ns-readme-map/ns-readme-build-53-5-con.html.

All enhancements, bug fixes, and known issues in this readme apply to Citrix NetScaler 9.3.e nCore™.

Note: A copy of this 9.3.e readme is also available in the Citrix Knowledge Center at http://support.citrix.com/.

Enhancements

Cloud Bridge

- Req ID 0270815: The configuration utility now includes a wizard that helps you to easily configure a cloud bridge between a NetScaler appliance on any network and NetScaler VPX instances on the SOFTLAYER enterprise cloud.

  For more information, see the Cloud Bridge chapter of the Citrix NetScaler Networking Guide, available at http://support.citrix.com/article/CTX130085.

- Req ID 0262566: The following statistical counters have been introduced for IPSEC tunnels:

  - Bytes Received
  - Bytes Sent
  - Packets Received
  - Packets Sent

  For more information, see the man page for the 'stat IPSEC counters' command.

- Req ID 0261540: For an IPSEC tunnel, the NetScaler appliance now performs the standard IKEv2 liveliness check on the peer at a regular interval, which is user configurable. Based on check, the NetScaler appliance displays the status of the tunnel as UP or DOWN.

- Req ID 0258969: By using the configuration utility, you can now configure a cloud bridge between a NetScaler appliance on any network and NetScaler VPX instances on the COTENDO enterprise cloud.

Networking

- Req ID 84792/0245142: Now, in a High Availability (HA) configuration, you can create route monitors in non-INC mode. Route monitors are propagated and get synchronized only in the non-INC mode. Route monitors are useful in a non-INC mode HA configuration where you want the non-reachability of a gateway from a primary node to be one of the conditions for HA failover.

  For more information, see the "Configuring Route Monitors" section in the High Availability chapter of the Citrix NetScaler Networking Guide, available at http://support.citrix.com/article/CTX130085.

- Req ID 0262405: You can now configure the NetScaler appliance to respond or not respond to ARP requests for a Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP.

  For more information, see the "Configuring ARP Response Supression for Virtual IP addresses (VIPs)" section in the IP Addressing chapter of the Citrix NetScaler Networking Guide, available at http://support.citrix.com/article/CTX130085.

- Req ID 94655/0258893: You can now bind a NetScaler owned SNIP address to an interface without using Layer 3 VLANs. Any packets related to the SNIP address will go only through the bound interface.

  For more information, see the "Binding an NetScaler Owned IP address to an Interface" section in the Interface chapter of the Citrix NetScaler Networking Guide, available at http://support.citrix.com/article/CTX130085.

Changes and Fixes

Networking Issues

- Issue IDs 94162/0257992 and 0262493: For a connection from a virtual server to the bound server, the NetScaler appliance uses the SNIP address instead of the net profile IPs configured for the virtual server.

Configuration Utility Issues

- Issue ID 0269486: In a High availability (HA) configuration, the configuration utility does not display the configured route monitors on the NetScaler appliances. Also, when a route monitor is configured to monitor a default route , the configuration utility displays the secondary node's IP address as 0.0.0.0.

System Issues

- Issue ID 93475/0257369: When the NetScaler appliance releases memory for a data structure, which is used for tracking NAT info, an associated field related to net bridge configuration fails to get released. When the server side connection picks the same data structure, the appliance sends data on a bridge, which is not configured. This leads to memory leak.
- Issue ID 94442/0258246 and 93593/0257475: When device name length exceeds 256 characters, then the length stored is truncated. However, the NetScaler appliance allocates more memory to store the device name and while releasing the memory, the appliance releases less memory than the extended. This leads to memory leak.

High Availability Issues

- Issue ID 92866/0251883: In an HA configuration in non-inc mode, on executing the config sync operation, the ‘bind Routemonitor’ command does not get synched to the secondary.
- Issue ID 93068/0252065: In an HA configuration in non-inc mode, on executing the "clear config" command or clearing config as a part of config sync operation, route monitors get modified, but health check is not done and thus node state is not updated properly.

Known Issues and Workarounds

Documentation Issues

- Issue ID 90875/0250110: On a TCP load balancing virtual server, if persistence is defined with the rule '`client.tcp.payload(n)`', and a request is received in multiple parts such that there is a delay between the parts and a FIN is sent from client before the expected number of bytes (n), the NetScaler appliance creates an undesired session with the received number of bytes (which is less than n). This behavior needs to be documented.

Global Server Load Balancing Issues

- Issue ID 93051/0252048: When a load balancing virtual server that is a part of a GSLB configuration is down and receives a client request, the NetScaler appliance makes a GSLB decision and attempts an HTTP redirect or a connection proxy to a GSLB site that is UP and healthy. The appliance fails when making this GSLB decision if source IP persistence is set on both the primary and backup GSLB virtual servers and if the core that receives the request is not the owner of the source IP persistence entry.

Load Balancing Issues

- Issue ID 90395/0249705: If the rule that is used for creating rule based persistence sessions is a compound expression, the "show lb persistentSessions" CLI command displays an internal representation of the persistence parameter instead of the actual persistence parameter.

- Issue ID 91672/0250820: If the token that is used for creating rule based persistence sessions is larger than 100 MB, the "show lb persistentSessions" CLI command does not display the persistence parameter.

- Issue ID 91711/0250846: If the string (or "token") that is used for creating rule based persistence sessions for load balancing virtual servers is larger than 64 KB, the NetScaler appliance fails to create persistence sessions. For example, the appliance fails to create persistence sessions with the rule `CLIENT.TCP.PAYLOAD(70000)` because the token that is used is larger than 64 KB. However, the appliance creates persistence sessions successfully with a rule such as `CLIENT.TCP.PAYLOAD(70000).BEFORE_STR("string2").AFTER_STR("string1")` if the string that is enclosed by "string1" and "string2" is not larger than 64 KB.

Build 51.5006.e

Release version: Citrix(R) NetScaler(R), version 9.3.e, Build 51.5006.e

Replaces build: None

Release date: September 2011

Language supported: English (US)

Readme version: 1.0

Review the following sections:

- Enhancements
- Known Issues and Workarounds

Note:

- All enhancements and known issues in this readme apply to Citrix NetScaler 9.3.e nCore™.
- For the list of bug fixes in this release, see the readme for Citrix NetScaler version 9.3 build 51.5.
- A copy of this 9.3.e readme is also available in the Citrix Knowledge Center at http://support.citrix.com.

Enhancements

Sessionless Field Consistency

With the sessionless Field Consistency Check feature, the application firewall does not store web forms in memory. Instead, it adds a hidden form field named as_ffc_field to each form before forwarding it to the client. When the client submits the form, the application firewall extracts as_ffc_field and compares it to the remaining form to establish field consistency.

By default, sessionless Field Consistency is disabled. The following CLI commands configure sessionless Field Consistency:

add appfw profile <name> -sessionlessFieldConsistency (ON|OFF|postOnly)

set appfw profile <name> -sessionlessFieldConsistency (ON|OFF|postOnly)

Virtual Server - Options of Response to PING

You can now configure the NetScaler not to respond to a ping message if the virtual server is DOWN. This is possible on load balancing, content switching, cache redirection, and VPN virtual servers. By default, the NetScaler responds to a ping message even if one or more virtual servers are DOWN. The option can be set at an IP address level or virtual server level. The option functions as described below:

On an IP address:

Option	Effect
NONE	Always responds
ONE_VSERVER	Responds if at least one virtual server on this IP address is UP
ALL_VSERVER	Responds only if all the virtual servers on this IP address are UP
VSVR_CNTRLD	Responds according to the setting on the virtual servers

On a virtual server:

PASSIVE on all virtual servers	Always responds
ACTIVE on all virtual servers	Responds even if one virtual server is UP
ACTIVE on some and PASSIVE on others	Responds even if one virtual server set to ACTIVE is UP

This option can be set on an IP address only if it is a VIP.

CLI commands:

set ip <IPAddress> -icmpresponse (NONE | ONE_VSERVER | ALL_VSERVERS | VSVR_CNTRLD)

set lb vserver <name> -icmpVsrResponse (PASSIVE | ACTIVE)

You can replace lb with cs, cr or vpn.

GUI:

Create/Configure IP dialog box: The ICMP Response dropdown list

Create/Configure Virtual Server>>Advanced tab: The ICMP VServer Response dropdown list

Denying Nonsecure SSL Renegotiation

SSL and TLS renegotiations are vulnerable to an MITM attack that injects its own content as a prefix to a TLS connection. A new option addresses this vulnerability. If you specify NONSECURE as the value of the denySSLReneg parameter in the "set ssl parameter" command, any nonsecure renegotiations are denied. For more information about this attack, see RFC 5746. For more information about setting this parameter, see "Configuring Advanced SSL Settings" in the "SSL Offload and Acceleration" chapter of the Traffic Management Guide at http://support.citrix.com/article/CTX130084.

Rule Based Persistence Support for Load Balancing Virtual Servers of Type TCP and SSL_TCP

You can now configure a rule to define persistence criteria for load balancing virtual servers of type TCP and SSL_TCP. The persistence criteria can be based on TCP/IP protocol data, Layer 2 data, TCP options, and TCP payloads (even if the protocol that is encapsulated in the TCP payload is not HTTP). In the "add lb vserver" or "set lb vserver" CLI command, set the "persistenceType" parameter to "RULE," and then configure a rule for the rule parameter. You can define rules to configure persistence based on source and destination ports, source and destination IP addresses and IP octets, source and destination MAC addresses, VLAN IDs, payload content, and so on. Following are examples of expressions that you can use to define persistence criteria:

- CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=',';').VALUE("field1"). The value of field1, obtained after casting the first 500 bytes of the TCP payload to a name-value list that consists of name-value pairs in the format <name>=<value>;.
- CLIENT.TCP.SRCPORT. The source port in the client request.
- CLIENT.IP.DST. The destination IP address in the client request.
- CLIENT.IP.SRC.GET4. The fourth octet (rightmost octet) of the source IP address in the client request.

- CLIENT.ETHER.DSTMAC.GET5. The fifth octet of the destination MAC address in the client request.

- CLIENT.VLAN.ID. The ID of the VLAN through which the request arrived.

Following is an example of a command that you can use to configure rule based persistence based on the destination IP address in the client request: add lb vserver mylbvserver SSL_TCP 192.0.2.0 443 -persistenceType RULE -rule CLIENT.IP.DST.

You cannot set the "resRule" parameter for load balancing virtual servers of type TCP and SSL_TCP.

For more information, see the "Load Balancing" chapter of the Citrix NetScaler Traffic Management Guide at http://support.citrix.com/article/CTX130084, which includes a section titled "Configuring Persistence Based on User-Defined Rules." The chapter also has a "Use Cases" section, in which "Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream" describes how to configure rule based persistence for servers that communicate Financial Information eXchange (FIX) protocol data over TCP.

Known Issues and Workarounds

Documentation Issue

- Issue ID 90875: On a TCP load balancing virtual server, if persistence is defined with the rule 'client.tcp.payload(n)', and a request is received in multiple parts such that there is a delay between the parts and a FIN is sent from client before the expected number of bytes (n), the NetScaler appliance creates an undesired session with the received number of bytes (which is less than n). This behavior needs to be documented.

Global Server Load Balancing Issue

- Issue ID 93051: When a load balancing virtual server that is a part of a GSLB configuration is down and receives a client request, the NetScaler appliance makes a GSLB decision and attempts an HTTP redirect or a connection proxy to a GSLB site that is UP and healthy. The appliance fails when making this GSLB decision if source IP persistence is set on both the primary and backup GSLB virtual servers and if the core that receives the request is not the owner of the source IP persistence entry.

Load Balancing Issues

- Issue ID 90395: If the rule that is used for creating rule based persistence sessions is a compound expression, the "show lb persistentSessions" CLI command displays an internal representation of the persistence parameter instead of the actual persistence parameter.

- Issue ID 91672: If the token that is used for creating rule based persistence sessions is larger than 100 MB, the "show lb persistentSessions" CLI command does not display the persistence parameter.

- Issue ID 91711: If the string (or "token") that is used for creating rule based persistence sessions for load balancing virtual servers is larger than 64 KB, the NetScaler appliance fails to create persistence sessions. For example, the appliance fails to create persistence sessions with the rule CLIENT.TCP.PAYLOAD(70000) because the token that is used is larger than 64 KB. However, the appliance creates persistence sessions successfully with a rule such as CLIENT.TCP.PAYLOAD(70000). BEFORE_STR("string2").AFTER_STR("string1") if the string that is enclosed by "string1" and "string2" is not larger than 64 KB.

Build 50.3002.e

Release version: Citrix(R) NetScaler(R), version 9.3.e, Build 50.3002.e

Replaces build: None

Release date: August 2011

Language supported: English (US)

Readme version: 1.0

Review the following sections:

Note:

- All enhancements and known issues in this readme apply to Citrix NetScaler 9.3.e nCore™.
- For the list of bug fixes in this release, see the readme for Citrix NetScaler version 9.3 build 50.3.
- A copy of this 9.3.e readme is also available in the Citrix Knowledge Center at http://support.citrix.com.

Enhancements

Content Switching

When you run the 'show cs vserver' command, you can now view the content switching policies associated with the virtual server in the order of the priority of the policies rather than by the chronological order in which they are bound.

This enhancement can help you know the order in which the content switching polices are applied and therefore, understand how client requests are routed. The configuration utility also shows the content switching policies in the order of their priority.

For more information, see the "Viewing the Properties of Content Switching Virtual Servers" section in the Content Switching chapter of the Citrix NetScaler Traffic Management Guide, available at http://support.citrix.com/article/CTX128670.

Networking

Now you can enable a NetScaler appliance to forward all the ICMP fragments of an ICMP echo request, destined to a network device, and the ICMP fragments of the corresponding echo response.

One of the examples where this enhancement is useful is a scenario including a NetScaler appliance and a Windows 2000 Server.

The Windows 2000 server sends out ICMP request of size 2048 for slow link detection. The NetScaler appliance successfully forwards the ICMP fragments of the ICMP request to the destined network device and the ICMP fragments of the ICMP response from the network device to the Windows 2000 server.

Surge Protection

If you want to flush the surge queue of a service, service group, or a load balancing or content switching virtual server, now you do not need to disable the NetScaler entity. With this enhancement, you can manage the traffic in surge conditions without affecting the existing traffic.

Options are added to the command line interface and configuration utility to flush a surge queue. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

When you flush the surge queue of a virtual server, the surge queues of all the services and service groups bound to it are flushed. When you flush the surge queue of a service group, surge queues of all its members are flushed. You can flush the surge queue of one or more members of a service group without flushing the surge queues of all its members. You can flush the surge queue of a specific service.

In the configuration utility, when you select an entity the 'Flush Surge Queue' option is available in the action pane. In the command line interface 'flush ns surgeQ' option is added with necessary options.

For more information, see the "Flushing the Surge Queue" section in the Load Balancing chapter of the Citrix NetScaler Traffic Management Guide, available at http://support.citrix.com/article/CTX128670.

Known Issues and Workarounds

High Availability Issues

- Issue ID 92866: In HA non-inc mode, on executing the config sync operation, the 'bind routemonitor' command is not getting synched to secondary.
- Issue ID 93068: In HA non-inc mode, on executing the 'clear config' command or clearing config as a part of config sync operation, routemonitors get modified, but health check is not done and thus node state is not updated properly.

Release Notes

These topics describe enhancements in NetScaler® 9.3 Classic, NetScaler® 9.3 nCore™, and NetScaler® 9.3 nCore™ VPX™ releases. The nCore NetScaler uses multiple CPU cores for packet handling, which greatly improves the performance of many NetScaler features.

Note: From this release, NetScaler® Classic VPX™ will not be available.

You can determine your NetScaler build type by looking at the build information in the upper-right corner of the NetScaler browser window, or by issuing the show version command at the command line. The file extension indicates the build type. In a browser, an nCore NetScaler has a .nc extension and a Classic NetScaler has a .cl extension. On the command line, the tar file name for an nCore NetScaler contains _nc and a Classic NetScaler contains _cl.

Note: Unless stated otherwise, the enhancements, known issues, and limitations apply to Citrix® NetScaler® 9.3 Classic, NetScaler 9.3 nCore™, and NetScaler® 9.3 nCore™ VPX™.

The NetScaler documentation (PDF format) for Citrix NetScaler release 9.3 is available on the **Documentation** tab of your NetScaler appliance. Most of the documents require Adobe Reader, available at http://adobe.com/.

To view the documentation for a given feature

1. Log on to the NetScaler appliance from a Web browser.
2. Click the **Documentation** tab.
3. To view a short description of each document, hover your cursor over the title. To open a document, click its title.

NetScaler 9.3 Enhancements

The following enhancements are available in this release.

Note: Unless stated otherwise, the enhancements apply to Citrix® NetScaler® 9.3 Classic, NetScaler 9.3 nCore™, and NetScaler® 9.3 nCore™ VPX™.

AAA

The following AAA feature enhancements are available in this release.

Kerberos Support for AAA

The NetScaler appliance can now authenticate a client/user by Kerberos or NTLM through Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) protocol. To configure this, define and bind a NEGOTIATE authentication policy. For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

AAA and Microsoft SharePoint

You can now successfully download or edit documents from Microsoft SharePoint sites after authenticating through AAA.

Access Gateway

The following Access Gateway enhancement is available in this release.

Idle Time-Out

If users log on with the Access Gateway Plug-in for Java and you configure an idle time-out on Access Gateway, the session does not end if the plug-in does not detect mouse or keyboard activity within the specified time limit.

Advanced Policies

The following policy enhancements are available in this release.

Virtual Server-Based Expressions

A new expression prefix, `SYS.VSERVER("<vserver-name>")`, enables you to identify a virtual server. You can use the `THROUGHPUT`, `CONNECTIONS`, `STATE`, `HEALTH`, `RESPTIME`, and `SURGECOUNT` functions with this prefix to retrieve information related to the specified virtual server. For more information about these functions, see the "Virtual Server Based Expressions" section in the "Default Syntax Expressions: Working with Dates, Times, and Numbers" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Named Expressions

You can now use the name of a default syntax expression as the prefix to a function. The named expression can be either a simple expression or a compound expression. The function must be one that can operate on the type of data that is returned by the named expression.

Example 1: Simple Advanced Expression

The following named expression, which identifies a text string, can be used as a prefix to the `AFTER_STR("<string>")` expression function, which works with text data:

```
HTTP.REQ.BODY(1000)
```

If the name of the expression is `top1000bytes`, you can use `top1000bytes.AFTER_STR("username")` instead of `HTTP.REQ.BODY(1000).AFTER_STR("username")`.

Example 2: Compound Advanced Expression

The name of the following compound named expression, which identifies a number, can be used in a comparison:

```
HTTP.REQ.HEADER("Header1").LENGTH + HTTP.REQ.HEADER("Header2").LENGTH
```

If the name of the compound expression is `headerlimit`, this enhancement allows you to use `headerlimit.GT(30)` instead of `(HTTP.REQ.HEADER("Header1").LENGTH + HTTP.REQ.HEADER("Header2").LENGTH) > 30`.

Additionally, you can use a named expression (either by itself or as a prefix to a function) to create the text expression for the replacement target in rewrite.

Support for Unsigned Long and Double Data Types

In NetScaler 9.3, simple expressions can return both double and unsigned long data types, thus allowing compound expressions that use arithmetic operators and logical operators to evaluate or return values of these data types. Additionally, you can use double and unsigned long values in policy expressions. For information about the functions that can work with the double and unsigned long data types, see the "Functions for Data Types in the Policy Infrastructure" section in the "Configuring Default Syntax Expressions: Getting Started" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Modifying (Corrupting) HTTP Headers

A new rewrite action, CORRUPT_HTTP_HEADER, replaces the name of the given HTTP header with a corrupted name so that it will not be recognized by the receiver.

Pattern Set Name Length

The name of a pattern set can now contain up to 127 characters.

Encryption and Decryption of Payloads

You can now configure the NetScaler appliance to encrypt and decrypt text and XML data in requests and responses. To encrypt and decrypt text, you use the ENCRYPT and DECRYPT functions, respectively. To encrypt and decrypt XML payloads, you use the XML_ENCRYPT() and XML_DECRYPT() functions, respectively. For more information about encrypting text, see the "Encrypting and Decrypting Text" section in the "Default Syntax Expressions: Evaluating Text" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673. For more information about encrypting XML payloads, see the "Encrypting and Decrypting XML Payloads" section in the "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Policy-Based Logging for Responder Policies

You can now log audit messages in a defined format when the rule in a responder policy evaluates to TRUE. To do this, you must configure an audit message action and associate it with a responder policy. The format of audit messages can be defined only with advanced expressions in an audit message. Audit message actions can be used to log messages at various log levels, either only in SYSLOG format or in both SYSLOG and newnslog formats. You can also define the SYSLOG messages to be logged. To log such SYSLOG messages, you must select **User Configurable Log Messages** (**System** > **Auditing** > **Change Global Auditing Settings**) in the NetScaler configuration utility or set the "UserDefinedLogging" SYSLOG parameter to YES in the NetScaler command line.

For example, you can configure a log action that defines the format of the log message and then associate the log action with a responder policy that can be bound globally or that can be bound to a load balancing or a content switching virtual server, as demonstrated in the following commands:

```
 Done
> set syslogparams -userDefinedAuditlog yes
 Done
> add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS Response Time to Servers:\" + sys
 Done
> add responder action redirect_url_2_act redirect "\"http://redirect_url.com\""
 Done
```

```
> add responder policy  redirect_policy_url_1  "http.req.url.eq(\"url_1.com\")" redirect_url_2_act RESET -log
 Done
> bind responder global redirect_policy_url_1 200 END -type REQ_DEFAULT
 Done
>
```

Support for the UTF-8 Character Set

The NetScaler policy infrastructure now supports the UTF-8 character set. For more information, see the “Specifying the Character Set in Expressions” section in the chapter, “Configuring Advanced Expressions: Getting Started,” in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Matching Text by Using a String Map

You can now use string maps to perform pattern matching in all NetScaler features that use the default policy syntax. A string map is a NetScaler entity that consists of key-value pairs. The keys and values are strings in either ASCII or UTF-8 format. String comparison uses two new functions, `MAP_STRING(<string_map_name>)` and `IS_STRINGMAP_KEY(<string_map_name>)`.

A policy configuration that uses string maps performs better than one that does string matching through policy expressions, and you need fewer policies to perform string matching with a large number of key-value pairs. String maps are also intuitive, simple to configure, and result in a smaller configuration.

For more information about string maps, see the "String Maps" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Rules for Names in Identifiers Used in Policies

The names of identifiers in the named expression, HTTP callout, pattern set, and rate limiting features must begin with an ASCII alphabet or an underscore (_). The remaining characters can be ASCII alphanumeric characters or underscores (_).

The names of these identifiers must not begin with the following reserved words:

- The words `ALT`, `TRUE`, and `FALSE` and the one-character identifiers `Q` and `S`.
- Words that indicate special syntax, which are `RE` (for regular expressions) and `XP` (for XPath expressions).
- Expression prefixes, which currently are the following:
  - CLIENT
  - EXTEND
  - HTTP

- SERVER
- SYS
- TARGET
- TEXT
- URL
- MYSQL

Additionally, the names of these identifiers cannot be the same as the names of enumeration constants used in the policy infrastructure. For example, the name of an identifier cannot be `IGNORECASE`, `YEAR`, or `LATIN2_CZECH_CS` (a MySQL character set).

Note: The NetScaler appliance performs a case-insensitive comparison of identifiers with these words and enumeration constants. For example, names of the identifiers cannot begin with `TRUE`, `True`, or `true`.

Stripping Characters From a String

You can use the `STRIP_CHARS(<string>)` operator to remove specific characters from the text that is returned by an advanced expression prefix. You can then use a text method on the resulting string. For example, you can compare the resulting string with the strings in a pattern set. You can also use the `STRIP_START_CHARS(<string>)` and `STRIP_END_CHARS(<string>)` functions to strip characters from the beginning and end of input strings, respectively. For more information about stripping characters from an input string, see the "Stripping Specific Characters from a String" section in the "Default Syntax Expressions: Evaluating Text" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Tool for Converting Classic Expressions to the Newer Default Expression Syntax

You can convert a classic expression to the default expression syntax by using the `nspepi` conversion tool. You can also use the tool to convert all the classic expressions in the NetScaler configuration to the default syntax (with the exception of NetScaler entities that currently support only classic expressions). For more information about converting classic expressions and NetScaler configuration files to the default syntax, see "Converting Classic Expressions to the Newer Default Expression Syntax" section in the "Introduction to Policies and Expressions" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Identifying a Response That is Associated with an HTTP Redirect

You can use the IS_REDIRECT function to determine whether a response is associated with a redirect. For information about the IS_REDIRECT function, see the "Expressions for HTTP Status Codes and Numeric HTTP Payload Data Other Than Dates" section in the "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Expressions for Generating the Day of the Week, as Strings, in Short and Long Formats

Two functions, WEEKDAY_STRING_SHORT and WEEKDAY_STRING, have been introduced for generating the day of the week as a string, in short and long formats, respectively. The strings that are returned are always in English. The prefix used with these functions must return the day of the week in integer format and the acceptable range for the value returned by the prefix is 0–6. Therefore, you can use any prefix that returns an integer in the acceptable range. An UNDEF condition is raised if the returned value is not in this range or if memory allocation fails.

Following are the descriptions of the functions:

Function	Description
<prefix>.WEEKDAY_STRING_SHORT	Returns the day of the week in short format. The short form is always 3 characters long with the first character in upper case and the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT returns Sun if the value returned by the WEEKDAY function is 0 and Sat if the value returned by the prefix is 6.
<prefix>.WEEKDAY_STRING	Returns the day of the week in long format. The first character in the long form is always in upper case and the remaining characters are in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING returns Sunday if the value returned by the WEEKDAY function is 0 and Saturday if the value returned by the prefix is 6.

Enhancements to Expressions That Process Binary Strings

The default policy infrastructure has had some changes and improvements regarding handling of binary strings. These are particularly useful for handling protocols that have binary values in them.

The two parameters of the GET_SIGNED16, GET_UNSIGNED16, and GET_SIGNED32 functions have been changed:

- The first parameter is now an offset, instead of an index into a list of identically sized items. This change enables the functions to handle items that are not aligned on the boundaries required by indexes.
  - For 16-bit values: instead of using `N`, use `2*N` for an aligned value.
  - For 32-bit values: instead of using `N`, use `4*N` for an aligned value.
- The second parameter now takes a mnemonic value instead of the nonintuitive `0` or `1`. The values are `LITTLE_ENDIAN` and `BIG_ENDIAN`.
  - What used to be `0` is now `LITTLE_ENDIAN`.
  - What used to be `1` is now `BIG_ENDIAN`.
- Expressions using these functions will need to be changed.

Example:

HTTP.REQ.BODY(100).GET_SIGNED16(7, BIG_ENDIAN)

Additionally, the following new functions have been introduced. The functions that produce binary strings, numbered 2 through 9 in the following list, are particularly useful in TCP rewrite, as replacement strings for binary data.

1. `<text>.GET_UNSIGNED32(n, endianness)`

   This function returns the 32-bit unsigned binary integer at offset `n`. If the offset would make part or all of the value outside of the current text, an `UNDEF` is raised. In the above expression:

   - `n` is the number of bytes from the current position to the first byte of the binary integer.
   - The “endianness” can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`.
   - The value returned is an unsigned long number. This is different from the other `GET` functions for binary strings, which return a number.

   Example:

   HTTP.REQ.BODY(100).GET_UNSIGNED32(30, LITTLE_ENDIAN)

2. `<number>.SIGNED8_STRING`

   This function produces an 8-bit signed binary string representing the number. If the value is out of range an `UNDEF` is raised. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STRING

3. `<number>.UNSIGNED8_STRING`

   This function produces an 8-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The value returned is of type text.

Example:

HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED8_STRING

4. `<number>.SIGNED16_STRING(endianness)`

   This function produces a 16-bit signed binary string representing the number. If the value is out of range, an `UNDEF` is raised. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)

5. `<number>.UNSIGNED16_STRING(endianness)`

   This function produces a 16-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LITTLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN

6. `<number>.SIGNED32_STRING(endianness)`

   This function produces a 32-bit signed binary string representing the number. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).AFTER_STR("delim").GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_

7. `<unsigned_long_number>.UNSIGNED8_STRING`

   This function produces an 8-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIGNED_LONG_AT.ADD(12).UNSIGNED8_STRING

8. `<unsigned_long_number>.UNSIGNED16_STRING(endianness)`

   This function produces a 16-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LITTLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UN

9. `<unsigned_long_number>.UNSIGNED32_STRING(endianness)`

   This function produces a 32-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The “endianness” can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).AFTER_STR("delim2").GET_UNSIGNED32(0, BIG_ENDIAN).ADD(2).UNSIGNED32_STRIN

Rewriting TCP Payloads

You can now treat the payload of a TCP packet as a raw stream of bytes and perform rewrite actions on the payload regardless of the protocol that the TCP connection is transmitting. You can evaluate TCP traffic that is associated with only those services that are of type `TCP` and `SSL_TCP`. You can configure policies that use the `CLIENT.TCP.PAYLOAD(<integer>)` expression prefix and the `SERVER.TCP.PAYLOAD(<integer>)` expression prefix to evaluate TCP and SSL_TCP traffic received from clients and servers, respectively. With these prefixes, you can use all types of existing string manipulation functions to identify the strings that you want to rewrite. To bring the policies into effect, you bind the TCP rewrite policies to load balancing virtual servers and content switching virtual servers of type `TCP` and `SSL_TCP`. You can also bind them to the following new bind points:

- OTHERTCP_REQ_DEFAULT
- OTHERTCP_REQ_OVERRIDE
- OTHERTCP_RES_DEFAULT
- OTHERTCP_RES_OVERRIDE

Note: The term "OTHERTCP" is used in the context of the NetScaler appliance to refer to all TCP or SSL_TCP requests and responses that you want to treat as a raw stream of bytes regardless of the protocol that the TCP packets encapsulate.

The order of evaluation of policies bound to these policy banks is the same as that for policies that evaluate HTTP traffic: TCP rewrite policies are evaluated from the most specific bind point to the most general bind point. That is, policies at the override bind points are evaluated first, followed by those at the virtual server bind points, and finally, those at the default bind points.

Note: A TCP rewrite policy does not evaluate connections that are active at the time at which it is bound to a bind point, but it evaluates the TCP connections that are initiated after binding is performed.

The following rewrite actions are supported in TCP rewrite policies.

- INSERT_AFTER
- INSERT_BEFORE

- REPLACE
- DELETE
- INSERT_AFTER_ALL
- INSERT_BEFORE_ALL
- REPLACE_ALL
- DELETE_ALL
- DROP
- RESET

Even though you cannot rewrite data other than that in TCP payloads, you can use all the expressions that are available for evaluating TCP traffic and information associated with lower network layers such as that in IP packets and Ethernet frames. Based on the information retrieved, you can perform an appropriate rewrite action, selected from the preceding list, on the TCP payload.

Additionally, the following two transform types (values for the `transform` parameter in the `add rewrite policyLabel` CLI command) have been introduced for rewrite policy labels:

- OTHERTCP_REQ
- OTHERTCP_RES

When you create a rewrite policy label, you specify `OTHERTCP_REQ` as the transform type to specify that the policies that are bound to the policy label are request-based policies. You can specify `OTHERTCP_RES` as the transform type to specify that the policies that are bound to the policy label are response-based policies.

Following are some limitations of the TCP rewrite feature:

- Named expressions and HTTP callouts are not supported in TCP rewrite policies.
- If the payload length received is less than the expected payload length, the NetScaler waits for the remaining bytes. Eventually, the request times out.

Example:

The following rewrite action `c_ip` identifies the first 150 bytes of the TCP payload and then inserts the string `CLIENT_IP` and the IP address of the client before the string `\r\n\r\n`.

add rewrite action c_ip INSERT_BEFORE "CLIENT.TCP.PAYLOAD(150).BEFORE_STR(\"\r\n\r\n\")" "\"CLIENT_IP\"

AppExpert

The following AppExpert enhancements are available in this release.

Deployment Files for AppExpert Applications

In earlier NetScaler releases, when you exported an AppExpert application, only a template file was created, in GZIP format. The template file contained only application-configuration information. You had to specify deployment-specific information, such as public endpoints and backend servers, in the **AppExpert Template Wizard** when importing the template file. With release 9.3, when you export an AppExpert application, a deployment file is created along with the application template file. Both files are created in XML format. The template file contains configuration-specific information and the deployment file contains deployment-specific information. In the **AppExpert Template Wizard**, you can specify use of the deployment file when you import the template file, or you can manually specify all the deployment information. If you specify the deployment file, the NetScaler appliance uses the deployment information in the deployment file. For more information, see the chapter, "AppExpert Applications and Templates," in the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Roll Back Support for AppExpert Application Import

In this release, if an error occurs during the import of an AppExpert application, the configuration changes that were made during the import process are automatically rolled back.

Creating Application Templates from Content Switching Virtual Servers

In earlier releases, you could export a content switching virtual server configuration to an application template only from the **Content Switching Visualizer**. Additionally, the template file was in GZIP format and included only configuration information. No deployment information was exported. In this release, you can export a content switching virtual server either from the **Content Switching Visualizer** or from the **Content Switching Virtual Servers** pane. A deployment file is created along with the template file. Both files are in XML format. For more information about exporting a content switching virtual server, template files, and deployment files, see the "Creating and Managing Template Files" chapter of the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Application Firewall

The following Application Firewall enhancements are available in this release.

Document Type Definitions (DTD) Support for XML Security Checks

Support has been added for document type definitions (DTDs) in the **Application Firewall Imports** pane, **XML Schema** tab. You can upload DTDs to the Application Firewall and they will be used for the relevant XML security checks.

Audit Server Configuration for Application Firewall

Support has been added to the NetScaler GUI for configuring the Application Firewall logs, in the **Auditing Policies** pane. In this pane, you can configure auditing to any server by adding that server in the **Servers** tab, and then adding an Audit policy that refers to that server. The process is the same as for AAA.

Learning Visualizer for Deployed Application Firewall Rules

The Application Firewall learning visualizer is now available for learned rules that you have already deployed, as well as rules that are pending review. For more information about how to use the learning visualizer, see the "Learning" section in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

SNMP Traps for Application Firewall Security Checks

SNMP traps have been added for each of the Application Firewall security checks. You configure the traps in the NetScaler configuration utility, under **System** > **SNMP** > **Traps**. For more information, see the *Citrix NetScaler SNMP OID Reference Guide* at http://support.citrix.com/article/CTX128676.

Application Firewall Custom Variables for HTML/XML Error Objects

You can now use five new variables to write troubleshooting information into an Application Firewall HTML or XML error object. These variables are:

- **${NS_TRANSACTION_ID}.** The transaction ID that the Application Firewall assigned to this transaction.
- **${NS_APPFW_SESSION_ID}.** The Application Firewall session ID.
- **${NS_APPFW_VIOLATION_CATEGORY}.** The specific Application Firewall security check or rule that was violated.
- **${NS_APPFW_VIOLATION_LOG}.** The detailed error message associated with the violation.

- **${COOKIE("<CookieName>")}.** The contents of the specified cookie. For `<CookieName>`, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.

To use these variables, you embed them in the HTML or XML of the error page object as if they were any ordinary text string. For more information, see the "Imports" chapter in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Cookie Encryption, Cookie Proxying, and Adding Flags to Cookies

The Application Firewall now supports encryption and proxying of cookies, and can be configured to add flags to cookies. For information about configuring this feature, see the "The Cookie Consistency Check" section in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Limiting the Number of Files Uploaded to a Web Site

By default, there is no limit on the number of files that can be uploaded to your protected Web site by using an HTML Web form or properly structured XML request. You can now limit the number of file uploads allowed on your protected Web sites. At the NetScaler command prompt, type:

set appfw profile -fileUploadMaxNum <integer>

For `<integer>`, substitute a number between 0 and 65535. If you set this value to 0, that disables all file uploads. If you set it to 65535, that allows unlimited uploads rather than stopping at 65535.

In the configuration utility, in the **Configure Application Firewall Profile** dialog box, **Settings** tab, you select the **Max File Uploads** check box, and enter a number in text box.

New Signatures Feature

All Application Firewall signatures, SQL injection and cross-site scripting patterns are now combined into a single XML-based signatures object file. You can download this file to your local computer and edit it in a text editor to add to or modify the lists of SQL injection keywords and special strings, SQL injection transformations, and cross-site scripting allowed and denied tags, attributes, and patterns. For more information, see the "Signatures" chapter in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Configuration Wizard Added to Application Firewall

The Application Firewall configuration wizard is now available to assist users in performing initial configuration of a new Application Firewall. You access the wizard on the main **Application Firewall** pane, in the details area, under **Getting Started**, by clicking **Application Firewall Wizard**.

Excluding XML Elements and Attributes from Security Checks

You can now exempt elements and attributes from the XML SQL Injection and Cross-Site scripting checks. To exempt an element from either check at the NetScaler command line, you issue one of the following commands:

bind appfw profile <name> -XMLSQLInjection <string> [-isRegex (REGEX | NOTREGEX)] [-location <location>

bind appfw profile <name> -XMLXSS <string> [-isRegex (REGEX | NOTREGEX)] [-location <location>]

For <name>, you substitute the name of the profile. For <string>, you substitute a literal string or a Posix-complaint regular expression that describes the element or elements to exempt. If you typed a regular expression, you must include " -isRegex REGEX". If you typed a string, this parameter is optional. For <location>, you type the location where this element is found in the XML document.

Inverse Regular Expressions

You can now easily create an inverse regular expression in the Application Firewall when you are adding a relaxation or rule to supported security checks. You do this by putting an exclamation point (!) at the beginning of the expression. You can use this new regular expression notation in Start URL, Form Field Consistency, Cookie Consistency, HTML SQL Injection, Cross-Site Request Forgery, or HTML Cross-Site Scripting check relaxations, or in Deny URL or Field Format check rules.

Note: You cannot use a solitary exclamation point to create an inverse expression. There must be a valid regular expression that follows the exclamation point, or it is treated as a text element and not as a logical inverse.

Adding XQuery Injection Patterns to XPath

The XPATH Injection settings can be used to detect XQuery violations, by adding the XQUERY injection pattern(s) that you want to block, and setting the injection type to XQUERY. In the configuration utility, you do this as follows:

1. In the navigation pane, expand **Application Firewall**, and select **Signatures**.
2. In the details pane, select the default Xpath Injection file, and then click **Add**.

3. In the **Add Signatures Object** dialog box, **Name** field, type a name for your new custom Xpath Injection file. You can also type a comment in the **Comment** field that describes the purpose of this custom file.

4. In the lower left-hand corner of the **Add Signatures Object** dialog box, click **Manage SQL/XSS patterns**.

5. In the **Manage SQL/XSS patterns** dialog box, **Filtered Results** window, select **Patterns**, and then in the button row beneath the **Filtered Results** window, click **Add**.

6. In the **Create Signatures Folder** dialog box, **Attributes** area, **delimiter** drop-down list, choose **Any**.

7. In the **type** text box, type **XQUERY**, and then click **OK**.

8. If your new folder is not selected, select it, and then click **Add**.

9. In the **Create Signatures Folder** dialog box, choose **keyword**, and then click **OK**.

10. Repeat steps 8 and 9, but choose **Special String** this time.

11. Add your XQUERY keywords and special strings by clicking the **Add** button beneath the details window below the **Filtered Results** window, filling out the **Create Signature Item** dialog box, and then clicking **OK**.

Web-Based GUI Editor for Imported Files

A GUI-based editor has been added for editing imported files in the **Application Firewall Import** pane. Users can now select a previously imported file, and then click **Open**, to open that file in a Web-based text editor, where they can make changes to the file. This allows minor modifications to be made without having to export the file to the local system for editing.

XML SOAP Array Rule Added to XDOS Check

The XML SOAP Array rule has been added to the XML denial-of-security (XDoS) check. This check protects SOAP processors on your XML server from attacks launched through SOAP Arrays that exceed either individual or total limits on size and dimensions. The check verifies the following attributes of SOAP arrays in requests:

- **XMLMaxSOAPArraySize.** Checks for SOAP arrays that exceed the configured maximum total SOAP Array size across the request.

  XMLMaxSOAPArrayRank. Checks for SOAP arrays that exceed the configured maximum individual SOAP Array dimensions, or rank.

You configure the XML SOAP Array rule under **Application Firewall** --> **Profiles**, by selecting a profile, and then clicking **Open**. In **the Configure Application Firewall Profiles** dialog box, you select **XML Denial of Service**, and then click **Open**. In the **Modify XML Denial of Service Check** dialog box, you select **Soap Array Check**, and enable or disable the check by using the appropriate buttons. For more information, see the *Citrix Application Firewall*

Guide at http://support.citrix.com/article/CTX128677, XML Security Checks chapter, XML Denial of Service Check section.

Application Firewall/XML Signature Updates

Users can update the signatures and patterns on their Application Firewall by importing an updated signatures object file under the same name as an existing signatures object. Signatures updates support the following:

- By default, when updating a signatures object file, the Application Firewall will display the local or network location from which a signatures object file was originally imported. The user can choose a different location, or accept the default.
- Signatures each have an identification number. By default, when updating, any signature in the update file that uses the same signature identification number as an existing signature replaces the existing signature. This means that any modifications that the user made to existing signatures by editing the existing file will be overwritten unless the user also modifies the identification number to a number that is not already in use. Any existing signatures that have unique IDs are preserved. Any signatures in the update file that have unique IDs are added to the signatures object file.
- Any configuration that the user performed in the configuration utility is preserved during an update.
- All in-memory profile configurations that use information in the signatures object are updated immediately, without restarting the NetScaler VPX process, the NetScaler appliance, or the Application Firewall appliance.
- The complete update process requires less than a minute.

For instructions on updating signatures, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677, Signatures chapter.

Importing Cenzic Scan Results as XML File

You can now import results from a Cenzic scan as a signatures object. To do so, you must save the results as an XSLT file, and then import them as a signatures object. During the import process, you select the “Convert External Format” check box and then converting the XSLT file during import. To import a Cenzic results file, and then browse to and select the XSLT file. The results are imported as a signatures object.

Compression

The following compression feature enhancements are available in this release.

Policy Manager for Compression

The NetScaler configuration utility now includes a policy manager for the HTTP Compression feature. By using the Compression Policy Manager, you can configure compression policies and/or bind them to various request-time and response-time bind points from a single dialog box.

To configure compression policy bindings by using the Compression Policy Manager

1. Click **HTTP Compression**, and then, in the details pane, click **Compression policy manager**.

2. In the **Compression Policy Manager** dialog box, click **Request** or **Response**, depending on whether you want the policy you are binding to be evaluated at request time or at response time.

   Click the name of the bind point to which you want to bind the policy. An example of a bind point is **Override Global**.

   The policies that have already been bound to the bind point appear in the dialog box.

4. To insert a policy, click **Insert Policy**, and then, in the drop-down list that appears in the **Policy Name** column, do one of the following:

   - Click **New Policy** to open the **Create Compression Policy** dialog box. In this dialog box, you can create a compression policy. After you create the policy in this dialog box, click **Create** to bind the policy to the bind point.

   - Click the name of an existing policy to bind it to the bind point.

5. To unbind a policy, click the name of the policy, and then click **Unbind Policy**.

6. To regenerate priorities, click **Regenerate Priorities**. The priority values are modified to begin at 100, with increments of 10, without affecting the order of evaluation.

7. To modify a policy that is already bound, click the name of the policy, and then click **Modify Policy**. If the policy is a built-in policy, the **View Compression Policy** dialog box displays the details of the policy. If the policy is a user-defined policy, the **Configure Compression Policy** dialog box appears and you can modify the policy.

8. After you have configured policies and policy bindings for the bind point, click **Apply Changes**.

9. Click **Close**.

Configuration Utility

The following configuration utility enhancements are available in this release.

EdgeSight Monitoring

The HTML Injection feature in NetScaler has been renamed to EdgeSight® Monitoring in the NetScaler Configuration Utility. You can now configure EdgeSight Monitoring for both load balancing and content switching virtual servers and configure the EdgeSight services. You can register the NetScaler appliance to an EdgeSight server and also access the EdgeSight server from the NetScaler.

Note: If you had configured the EdgeSight Monitoring feature on the NetScaler appliance before upgrading to release 9.3, you must configure the feature again on the virtual servers.

IP Address Type Identification in VLAN Configuration Dialog Boxes

A new column, Type, has been added to the table in the IPs section of the Create VLAN and Configure VLAN dialog boxes. This column displays the IP address type (such as mapped IP, virtual IP, and subnet IP) for each IP address in the IP Addresses column.

Default Policy Format for NetScaler Features

Starting with NetScaler release 9.3, the newer advanced policy syntax is referred to as the "default syntax." Additionally, the policy configuration dialog boxes for NetScaler features that support both the older classic policy syntax and the default syntax no longer include the Classic Syntax and Advanced Syntax option buttons. By default, the policy configuration dialog boxes accept policy expressions that use the default syntax. The option buttons have been replaced by a hyperlink that enables you to switch between the classic syntax and the default syntax.

Viewing SYSLOG Event IDs

You can now view the unique identification number of every NetScaler SYSLOG event in the Syslog Viewer dialog box.

IP Address Range Support

Now, as an alternative to creating IP addresses one at a time, you can specify a consecutive range of IP addresses.

Customized System User Prompt

You can also customize the NetScaler command-line prompt for a user. Prompts can be defined in a user’s configuration, in a user-group configuration, and in the global configuration. The prompt displayed for a given user is determined by the following order of precedence:

1. Display the prompt as defined in the user's configuration.
2. Display the prompt as defined in the group configuration for the user’s group.
3. Display the prompt as defined in the system global configuration.

UTF-8 Support in Regular Expression Editors

The regular expression editor in the configuration utility now supports the UTF-8 character set and parses and displays details of regular expressions that consist of UTF-8 characters.

Documentation

The following documentation enhancements are available in this release.

NetScaler Documentation on eDocs

The 9.3 documentation suite is now available on Citrix® eDocs, the documentation portal. PDF versions of the guides will continue to be available on http://support.citrix.com and Configuration Utility (the NetScaler® graphical user interface). The salient features of the NetScaler® documentation library on eDocs are as follows:

1. **Intuitive organization.** Information is presented in a flatter and more intuitive manner. Each main node represents a release, and the subnodes represent specific features or tasks. In addition, the first few nodes cover basic configuration tasks to help new users adopt the product quickly.
2. **Global search.** Users can now search across releases and features. Users can also create their own search scopes.
3. **PDF on-the-fly.** Users can custom-create their own PDF documents by selecting information from specific nodes in eDocs.

Enhanced Context-Sensitive Online Help

The context-sensitive online help has been enhanced to display links to blogs, videos, articles, and documentation pertaining to the dialog box or pane from which the online help is invoked. This is made possible by an intelligent content aggregator that fetches results from key Citrix sites such as Knowledge Center, Citrix TV, and eDocs. The new help system is compatible with most popular Web browsers, such as Internet Explorer, Chrome, Safari, and Firefox.

Hardware

The following hardware enhancement is available in this release.

Support for 1G SFP Hot Swap

This release introduces support for SFP hot swap on the Intel 1G (e1k) driver for both fiber and copper SFPs, on the following NetScaler hardware platforms:

- MPX 7500/9500
- MPX 9700/10500/12500/15500
- MPX 9700/10500/12500/15500 10G and 10G FIPS

High Availability

The following high availability enhancements are available in this release.

Synchronizing Configuration Files in High Availability Setup

In a high availability (HA) setup, the `sync HA files` command now supports the following two new synchronization modes:

`sync misc`. Synchronizes all licenses and the configuration (rc.conf) file.

`Sync_all_plus_misc`. Synchronizes all files, including licenses and rc.conf file.

Enhanced Force Failover Warning Message

In a high availability configuration, the warning message for the force failover command now also includes the information the warning is based on.

HTML Injection (EdgeSight Monitoring)

The following HTML Injection enhancement is available in this release.

Details of Virtual Server and Service

When you enable HTML injection, you can view the name, IP address, and port number of the virtual server that load balanced the request and of the physical service that sent the response to the request.

Integrated Cache

The following integrated cache enhancements are available in this release.

Integrated Cache Support for Single Byte-Range Requests

The integrated cache can now serve partial content in response to single byte-range requests.

Built-in Content Group, Pattern Set, and Policies for the Integrated Cache

Release 9.3 includes a built-in integrated caching configuration that you can use for caching content. The configuration consists of a content group called ctx_cg_poc, a pattern set called ctx_file_extensions, and a set of integrated cache policies. In the content group ctx_cg_poc, only objects that are 500 KB or smaller are cached. The content is cached for 86000 seconds, and the memory limit for the content group is 512 MB. The pattern set is an indexed array of common file extensions for file-type matching. For more information about the built-in configuration, see the "Built-in Content Group, Pattern Set, and Policies for the Integrated Cache" section in the "Integrated Caching" chapter of the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Monitoring Utility

The following monitoring utility enhancement is available in this release.

Viewing Dashboard Functionalities on the Monitoring Utility

You can now view the dashboard functionalities in HTML on the Monitoring interface. You can no longer view the Java Dashboard on nCore version.

The HTML Dashboard feature provides the following new functionalities:

- **Comparative charts**:
  - The two comparative charts on the left side of the **Monitoring** page help in monitoring a built-in set of counters for all the global and entity-based system statistics.
  - The charts are auto-updated and depict the variation of two or more counters over time. You can view the related historical data by clicking the **Reporting** link.
  - Maximizing the charts opens the charts in a new window. You can also add or delete counters from this view.
  - In addition to the built-in comparative charts, you can also load, save, or delete custom charts.
- **Graphical view**:
  - You can view the performance statistics data graphically by clicking **Graphical View**.

- You can hide or unhide the counter plot by clicking the legend name.
- By default, the graphical view for multiple-entity stat commands is displayed for the entity that has the maximum value for a predefined counter. You can, however, choose a different entity by clicking **Counters** and view it graphically.
- You can add or remove counters and other entities by clicking **Counters** on the bottom right corner of the page.
- Multiple counters for multiple entities with a maximum of 16 data series can be plotted.
- The graph also displays historical data for a maximum period of five minutes.

- **Virtual server view**: You can view the statistics of multiple load balancing virtual servers in consecutive rows, which have fixed headers, with all the statistic attributes lined up in columns. This view is also applicable to other entities, such as GSLB virtual servers, services, and interfaces.

Gaps between Java Dashboard and HTML Dashboard: The following features which were supported on Java Dashboard are currently not supported on HTML Dashboard:

- Viewing the per-core usage and per-interface throughput charts, which can be obtained by clicking the **CPU Usage** and **InUse Memory** meters
- Customizing the chart type
- Viewing charts for bound entities
- Viewing the graphs displayed in the graphical view in an independent window
- Changing the units for the counters on the Y axis from Rate to Total and the other way round

Domain Name System (DNS)

The following Domain Name System enhancements are available in this release.

DNS Security Extensions on the NetScaler

You can configure DNS Security Extensions (DNSSEC) on the Citrix® NetScaler® appliance. You can configure DNSSEC for zones for which the NetScaler appliance is authoritative. You can configure the NetScaler appliance as a DNS proxy server for signed zones hosted on a farm of backend name servers. If the NetScaler appliance is authoritative for a subset of the records belonging to a zone for which the appliance is configured as a DNS proxy server, you can include the subset of records in the DNSSEC implementation. For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Controlling the Number of Pipeline Requests on an Individual Client Connection

You can now control the number of pipelined DNS requests on a single client connection, which is identified by the <clientip:port>-<vserverip:port> tuple.

NetScaler VPX

The following NetScaler VPX enhancements are available in this release.

3 Gbps Throughput on the NetScaler Virtual Appliance (nCore VPX only)

The Citrix NetScaler VPX virtual appliance now supports a throughput of 3 gigabits per second (Gbps), which requires a 3 Gbps license on the virtual appliance.

VMTools Support (nCore VPX only)

VMTools provides the ability to perform graceful shutdown and reset of the virtual appliances by using the vSphere/vCenter clients. VMTools now reports the version of VMTools running, and also the state of each virtual appliance, to the ESX server. You can also use VMTools to directly enter the IP address, gateway, and subnet mask of each virtual appliance without starting the virtual appliance.

VLAN Tagging (nCore VPX only)

The NetScaler virtual appliance hosted on VMware ESX or ESXi platform now supports VLAN tagging.

Networking

The following networking enhancements are available in this release.

RNAT NAT IP Range

When creating a RNAT entry using the NetScaler command line, you can now specify a range of IP addresses for the **natip** parameter. RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified.

PBR with Multiple Next Hops

Instead of sending the selected packets to a next hop router, you can now configure the PBR to send them to a link load balancing (LLB) virtual server to which you have bound multiple next hops. So if one of the next hop fails, all the packets that matched against a PBR are routed to other bound next hops as determined by the LB method configured on the LLB virtual server.

Round Robin Method for Selecting Source IPs for IP Tunnels

Now you can set an IP tunnel global parameter that enables the NetScaler appliance to use a different source IP address for each new session through a particular IP tunnel, as determined by round robin selection of one of the SNIP addresses. This setting is ignored if a common global source IP address has been specified for all the IP tunnels. This setting does not apply to a tunnel for which a source IP address has been specified.

Simple ACL6

Now you can quickly create ACLs, called simple ACL6s that filter IPv6 packets on the basis of their source IP address and, optionally, their destination port and/or their protocol. Any packet that has the characteristics specified in the ACL is dropped.

Inbound Network Address Translation (INAT)

The following Inbound Network Address Translation (INAT) configurations are now supported:

- **IPv4-IPv6 Mapping:** A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance creates an IPv6 request packet with the IP address of the IPv6 server as the destination IP address.
- **IPv6-IPv4 Mapping:** A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance creates an IPv4 request packet with the IP address of the IPv4 server as the destination IP address.
- **IPv6-IPv6 Mapping:** A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.

Prefix-Based IPv6-IPv4 Translation

Now the NetScaler appliance can translate packets sent from private IPv6 servers into IPv4 packets, using an IPv6 prefix configured in the NetScaler appliance. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix.

The NetScaler appliance compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix. If there is a match, the NetScaler appliance generates an IPv4 packet and sets the destination IP address as the last 32 bits of the destination IP address of the matched IPv6 packet.

IPv6 packets addressed to this prefix have to be routed to the NetScaler so that the IPv6-IPv4 translation is done by the NetScaler.

Pinging Link Local Address from Any VLAN

An option that lets you specify a VLAN ID has been introduced in the `Ping6` command for pinging a link local address from the VLAN.

Host Name-Based SNMP Managers

While adding an SNMP manager, you can now specify an associated host name of the SNMP manager instead of its IP address. To do so, you need to add a DNS name server that resolves the host name of the SNMP manager to its IP address. You can add up to a maximum of five host name-based SNMP managers in addition to hundred IP address-based managers.

Time-out for Dynamic ARP Entries

You can now globally set an aging time (time-out value) for dynamically learned ARP entries. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time. You can specify an ARP time-out value from 1 through 1200 seconds.

Alias Name for a VLAN

An option for adding alias names for VLANs has been introduced, with a view to enhancing readability. However, you cannot perform VLAN operations by specifying alias names instead of VLAN IDs.

Option for Tagging or Untagging the NSVLAN

An option for tagging or untagging the NSVLAN (the VLAN to which NSIP address is bound) has been introduced. By default, the NSVLAN is tagged.

New vtysh Commands

The following vtysh commands have been introduced:

- **`show ns ha sync-status`:** Displays ZebOS config sync status.
- **`ns ha sync`:** Synchronizes ZebOS routing configuration between the primary and secondary nodes in a high availability configuration.

NITRO API

The following NITRO API enhancements are available in this release.

NITRO API Support for AppExpert Applications

NITRO API support has been added for exporting and importing of AppExpert applications.

Exporting an AppExpert application. When you export an AppExpert application, the following two XML files are automatically created:

- **Application template file**. Contains the AppExpert application's configuration information, such as application units, rules, and configured policies. The template file is stored in the /nsconfig/nstemplates/applications directory on the NetScaler appliance.
- **Deployment file**. Contains deployment-specific information of the AppExpert application, such as public endpoints, services, and configured variables. By default, the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory on the NetScaler appliance.

Importing an AppExpert application. You can import an AppExpert application template file, or a template file and its corresponding deployment file, to a NetScaler appliance. If you import both files, all of the application's configuration information is imported from the template file and all deployment-specific information is imported from the deployment file. Alternatively, if you import just the template file, you can set parameters to configure your own deployment settings (such as public endpoints (VIPs), services, and variable values).

NITRO API Support for Entity Templates

You can now import and export entity templates for a load balancing virtual server through the NITRO API version of the NetScaler appliance. The templates are saved as XML files.

Secure Sockets Layer (SSL)

The following Secure Sockets Layer enhancements are available in this release.

SHA-2 Signature Algorithm Support

The NetScaler appliance now supports client certificate authentication for certificates signed with the SHA-2 signature algorithm.

Small Records Processing

You can now insert the PUSH flag into decrypted, encrypted, or all records. If the PUSH flag is set to a value other than 0, the buffered records are forwarded to the server on the basis of the value of the PUSH flag.

The SSL engine has been enhanced to better handle small records received from a client or a server. This change optimizes the handling of multiple small SSL records in one TCP packet. Together with the option to set the PUSH flag after encryption or decryption as described above, the new feature helps prevent the generation of small packets on the server or the client.

Server Name Indication

If Server Name Indication (SNI) is enabled on the virtual server, all client requests in which the client initiates the handshake for one domain and sends an HTTP request for another domain are now dropped by the NetScaler appliance.

Support for IPv6 Addresses in Online Certificate Status Protocol

The NetScaler appliance now supports IPv6 addresses in Online Certificate Status protocol (OCSP) for certificate management.

Inserting Complete Client Certificate in Online Certificate Status Protocol

The NetScaler appliance now inserts the complete client certificate in its query for certificate status to the OCSP responder.

Advanced Encryption Standard New Instructions

Advanced Encryption Standard (AES) New Instructions (AES-NI) is now supported on the NetScaler VPX virtual appliance. AES-NI is supported on systems with Intel Westmere processors. On appliances with this processor, SSL bulk encryption performance for AES cipher will be significantly enhanced.

PUSH Flag-Based Encryption Trigger Mechanism

The encryption trigger mechanism that is based on the PUSH TCP flag now enables you to do the following:

- Merge consecutive packets in which the PUSH flag is set into a single SSL record or ignore the PUSH flag on packets.
- Perform timer-based encryption, in which the time-out value is set globally using the `set ssl parameter -pushEncTriggerTimeout <positive_integer>` command.

The options that are available for the PUSH encryption trigger enhancements are:

- **Always.** Any PUSH packet triggers encryption
- **Ignore.** Ignore PUSH packet for triggering encryption
- **Merge.** For consecutive sequence of PUSH packets, last PUSH packet triggers encryption
- **Timer.** PUSH packet triggering encryption delayed by timer period defined in '`set ssl parameter`'

The default value is "Always."

Importing an External Key as a FIPS Key (nCore MPX FIPS appliance only)

You can now import an external key as a FIPS key on the MPX 9700/10500/12500/15500 10G FIPS appliances. The key should be in PEM format and not be encrypted. When importing an external key on the MPX appliances, you do not need a wrap key, nor do you have to convert the key to PKCS8 format.

SSL Enhancements in the NetScaler Configuration Utility

Configuring the NetScaler for SSL Offloading has been made simpler with the following enhancements to the SSL landing page in the NetScaler configuration utility:

- Direct links to create a server certificate, root CA certificate, intermediate CA certificate, and client certificate.
- Access to the OpenSSL interface from the configuration utility.

In addition:

- When you remove a certificate-key pair, you are prompted to remove the certificate file from the appliance.
- To avoid errors in linking certificates, when you select a certificate and click Link, only certificates that can be linked to the selected certificate are listed.

Warm Restart Option On 10G FIPS Appliances (nCore only)

Warm restart is now supported on MPX 9700/10500/12500/15500 10G FIPS appliances.

System

The following system enhancements are available in this release.

Enhanced Application Visibility Using AppFlow (nCore and nCore VPX only)

The Citrix® NetScaler® appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called flow records, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

Rate Limiting Notification (nCore and nCore VPX only)

You can now configure a threshold to trigger an alarm to send SNMP traps when the throughput rate (in Mbps) reaches the specified threshold. You must configure the threshold to a value lesser than the maximum throughput of the NetScaler appliance. The maximum throughput is based on the license installed on the NetScaler appliance. The platform sends the platformRateLimitThresholdHigh trap when the throughput rate exceeds the high threshold. When the throughput rate returns to the normal limit, the platformRateLimitThresholdNormal trap is sent.

You can use the following command to set the snmp alarm and specify the maximum and the normal threshold limits:

set snmp alarm PF-RL-RATE-THRESHOLD [**-thresholdValue** <positive_integer> [**-normalValue** <positive_integer>]] [**-time** <secs>] [**-state** (**ENABLED** | **DISABLED**)] [**-severity** <severity>] [**-logging** (**ENABLED** | **DISABLED**)]

Optimizing the TCP Maximum Segment Size for a Virtual Server Configuration

In earlier releases, when a client connects to a virtual server on the NetScaler appliance, the appliance responds with either the default Maximum Segment Size (MSS) value of 1460 or the MSS advertised by the client. Now, you can specify the MSS that the appliance advertises to a client when the client initiates a connection to a virtual server on the appliance. You can configure the MSS for the virtual servers configured on the appliance in two ways:

- You can set the MSS for each virtual server to a value of your choice in a TCP profile.
- You can set the learnVsvrMSS global TCP parameter to ENABLED to enable MSS learning for all the virtual servers configured on the appliance.

For more information about optimizing the TCP maximum segment size for a virtual server configuration, see the "Advanced Configurations" chapter of the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Specifying a TCP Buffer Size Globally and for Virtual Servers and Services

Now, you can set the TCP buffer size both globally and for individual virtual servers and services through TCP profiles. The value that you set is the minimum value that is advertised by the NetScaler appliance and reserved when clients initiate a connection that is associated with an endpoint application function such as compression or SSL. If the TCP buffer size is set both at the global level and at the entity level (virtual server or service level), the buffer size that is specified for the virtual server takes precedence over the global value. For more information about specifying a TCP buffer size globally and for virtual servers and services, see the "Advanced Configurations" chapter of the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Memory Consumption Statistics for NetScaler Features

You can run the following command at the command prompt to view the NetScaler memory currently allocated for a particular feature, the percentage of memory allocated for the feature, and the total number of allocation failures for the feature:

```
stat ns memory
```

Statistics for Overall System Memory Consumption

You can view statistics for overall system memory consumption by running the following command at the NetScaler command prompt:

```
Stat system memory
```

The output of the command includes information about total available system memory, memory allocated in megabytes, percentage of memory allocated, in-use memory in megabytes, percentage of memory in use, and shared memory.

Filtering a NetScaler Trace to Capture Information About an Interface or a VLAN ID

You can now run a trace on the NetScaler appliance and filter packets associated with a given interface or VLAN ID. Only the packets that are filtered are captured in the trace files. To filter the trace for a given interface or VLAN ID, you must include the filter option in the nstrace command. The option takes a simple or compound expression. The expression must specify the interface or VLAN ID for which the trace must be filtered.

To capture the trace for a given interface on the appliance, at the NetScaler command prompt, type the following command:

nstrace -filter <expression>

The following example captures the trace at interface 1/1 by using a simple expression:

```
nstrace -filter "intf == 1/1"
```

The following example captures the trace at interface 1/1 and further filters the trace for packets associated with TCP destination port 80:

```
nstrace -filter "intf == 1/1 && destport == 80"
```

To capture the trace for a given VLAN ID, at the NetScaler command prompt, type the following command:

nstrace -filter "vlan <operator> <vlanid>**"**

The operator in the expression for filtering the trace on the basis of a VLAN ID can be equal to (==), not equal to (!=), less than (<), greater than (>), less than or equal to (<=), or greater than or equal to (>=).

The following example captures the trace for a VLAN with ID equal to 1:

```
nstrace -filter "vlan == 1"
```

Time Stamp of the Last Configuration Change

The “show nsconfig” command now displays the time stamp of the most recent change in the running configuration on the NetScaler.

View System Date in the NetScaler Command Line

The show ns config now displays the system date and time. The date and time of the NetScaler appliance is the date and time of the location in which the NetScaler is located. To change the system date and time, you must use the shell interface to the underlying FreeBSD OS.

Omitting Device-Specific Information in Output

An option `ignoredevspecific` has been introduced in the `diff ns config` command for not displaying device-specific information (such as IP address and host name) in the output.

Web Interface on NetScaler (nCore and nCore VPX only)

The Web Interface on NetScaler is based on Java Server Pages (JSP) technology and provides access to Citrix® XenApp™ and Citrix® XenDesktop® applications. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in.

The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance. The Web Interface sites provide user access to the XenApp and XenDesktop resources, which include applications, content, and desktops.

For information about the Web Interface feature, see the "Advanced Configuration" chapter of the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Traffic Management

The following traffic management enhancements are available in this release.

Load Balancing of Branch Repeaters for WAN Optimization (nCore and nCore VPX only)

Now, to provide high-scale WAN optimization, the branch repeaters deployed at data centers can be load balanced through NetScaler appliances. The bandwidth and number of concurrent sessions can be improved significantly.

DataStream™ (nCore and nCore VPX only)

The DataStream feature of NetScaler provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a decision based on SQL query parameters, such as user name and database name, command parameters. You can further configure monitors to track the state of database servers.

The advanced policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with `MYSQL.CLIENT` and `MYSQL.REQ`) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with `MYSQL.RES`) to evaluate server responses to user-configured health monitors. For more information about these expressions, see the "Evaluating Connections to Database Servers" chapter in the *Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: DataStream is supported only for MySQL databases. For information about the DataStream feature, see the "DataStream" chapter in the *Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

XenApp and XenDesktop Load Balancing Wizard Enhancements

Now, through the wizards for load balancing Citrix XenApp and Citrix XenDesktop, you can do the following:

- Choose to use the HTTP protocol for the backend communication even when the protocol for front-end communication is SSL.
- Specify any port number or use the port number specified on the virtual server.
- By default, the wizard selects the site path for the latest version of WI 5.x. You can select any path or specify a path of your choice.

No-Monitor Option for Services and Service Groups

If you do not want the NetScaler appliance to monitor a service or service group, you can disable monitoring by using the `healthMonitor` option for that service or service group. When the healthMonitor option is set to `NO`, the NetScaler appliance does not monitor the health of the service, and the service is always shown as `UP`.

Hash-Based Load Balancing Method

The hash-based load balancing algorithms have been enhanced to support pseudo-group persistency across multiple virtual servers. This pseudo-group persistency can be achieved by specifying a hash ID, which will be used for calculating the hash.

Support for Remote Desktop Protocol Load Balancing (nCore and nCore VPX only)

Starting this release, the NetScaler appliance supports Remote Desktop Protocol load balancing with session persistency. If the user reconnects with the session cookie, the NetScaler appliance connects the client to the server to which it was previously connected.

Policy-Based Routing Domains (nCore and nCore VPX only)

Now, enabling the L2Conn parameter for a load balancing virtual server allows multiple TCP and non-TCP connections with the same 4-tuple (<source IP>:<source port>::<destination IP>:< destination port>) to co-exist on the NetScaler appliance. The appliance uses both the 4-tuple and the Layer 2 parameters to identify TCP and non-TCP connections. The L2Conn parameters are the MAC address, VLAN ID, and channel ID, which are the Layer 2 parameters.

You can enable the L2Conn option in the following scenario:

1. Multiple VLANs are configured on the NetScaler appliance, and a firewall has been set up for each VLAN.

2. You want the traffic originating from the servers in one VLAN and bound for a virtual server in another VLAN to pass through the firewalls configured for both VLANs.

Note: In NetScaler 9.3 Classic, the l2Conn parameter, which specifies that Layer 2 connection parameters must also be used to uniquely identify a connection, is not supported for load balancing virtual servers even though it is listed in the CLI command synopses for load balancing virtual servers. The parameter is supported only for cache redirection virtual servers. In NetScaler 9.3 nCore, the parameter is supported for both load balancing virtual servers and cache redirection virtual servers. Therefore, when an nCore NetScaler appliance on which the l2Conn parameter is set for one or more load balancing virtual servers is downgraded to a Classic build or to an nCore build that does not support the l2Conn parameter, the load balancing configurations that use the l2Conn parameters are lost.

Graceful Shutdown of Backend Services

When you disable a service, if you cannot estimate the approximate amount of time it takes for all the connections of the service to complete their transactions, you can choose to shut down the service gracefully. In case of a graceful shutdown, the service is moved out of the service state only after all the clients connected to that service have either completed their transactions or are closed.

Additional Statistics for Load Balancing Virtual Servers

Now, in the statistics of virtual servers, you can see at the same time the number of hits per second for all the virtual servers configured on the NetScaler.

Persistence Time-out Option

The default time-out value for any persistence other than the cookie persistence is two minutes. If the time-out value is set to less than two minutes, the following error message appears:

ERROR: "Timeout value out of range; enter a value between 2 minutes and 1440 minutes"

Virtual Servers Bound to a Service Group

A new command show servicegroupbindings <serviceGroupName> displays the load balancing virtual servers that are bound to a service group.

Domain-Name-Based Service Groups

You can now bind domain-name-based service (DBS) members to a service group, in addition to IP-address-based members. If you bind the member on the basis of its domain name, you need not reconfigure the member on the NetScaler whenever the IP address of the member changes. The NetScaler automatically detects such changes.

N-tier Cache Redirection

You can deploy NetScaler appliances in two tiers (layers), with the appliances in the upper tier load balancing those in the lower tier, and the appliances in the lower tier load balancing the cache servers. The two-tier cache redirection helps in handling huge amounts (several Gbps) of traffic.

Sessionless Load Balancing in IP Mode

You can do sessionless load balancing in the IP-based forwarding mode as well as in the MAC-based forwarding mode. For sessionless load balancing in the IP mode, you need not configure the IP address of the virtual server on the physical servers.

Customization of the HTTPONLY Flag (Classic only)

You can now customize the addition of the “httponly” flag in persistence cookies. At the NetScaler command line, type:

set lb parameter [**-httpOnlyCookieFlag** (ENABLED|DISABLED)]

The flag is enabled by default.

Enhancement of XenDesktop Monitors

The XD-DDC monitor for the Dynamic Desktop Controller (DDC) servers of Citrix® XenDesktop™ includes more intelligent health checks. The DDC monitor can validate the login credentials.

The new CITRIX-WEB-INTERFACE monitor for the Web Interface component monitors a dynamic page in the specified site path. The dynamic page helps to check for critical failures in resource availability.

You can use the CITRIX-WI-EXTENDED monitor to verify the validity of the login credentials, correct configuration of the monitor (for example, the site path), and the connection with the IIS server.

Link Load Balancing

You can now configure multiple link load balancing (LLB) routes by specifying the same virtual server name as the gateway.

Rule-Based Persistence for ANY Type Virtual Servers

You can configure the load balancing virtual servers that have the service type as "ANY" with rule-based persistence. This can be used for load balancing the Branch Repeater appliances for WAN optimization and for any other physical servers.

NetScaler 9.3 Enhancements

The following enhancements are available in this release.

Note: Unless stated otherwise, the enhancements apply to Citrix® NetScaler® 9.3 Classic, NetScaler 9.3 nCore™, and NetScaler® 9.3 nCore™ VPX™.

AAA

The following AAA feature enhancements are available in this release.

Kerberos Support for AAA

The NetScaler appliance can now authenticate a client/user by Kerberos or NTLM through Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) protocol. To configure this, define and bind a NEGOTIATE authentication policy. For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

AAA and Microsoft SharePoint

You can now successfully download or edit documents from Microsoft SharePoint sites after authenticating through AAA.

Access Gateway

The following Access Gateway enhancement is available in this release.

Idle Time-Out

If users log on with the Access Gateway Plug-in for Java and you configure an idle time-out on Access Gateway, the session does not end if the plug-in does not detect mouse or keyboard activity within the specified time limit.

Advanced Policies

The following policy enhancements are available in this release.

Virtual Server-Based Expressions

A new expression prefix, `SYS.VSERVER("<vserver-name>")`, enables you to identify a virtual server. You can use the `THROUGHPUT`, `CONNECTIONS`, `STATE`, `HEALTH`, `RESPTIME`, and `SURGECOUNT` functions with this prefix to retrieve information related to the specified virtual server. For more information about these functions, see the "Virtual Server Based Expressions" section in the "Default Syntax Expressions: Working with Dates, Times, and Numbers" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Named Expressions

You can now use the name of a default syntax expression as the prefix to a function. The named expression can be either a simple expression or a compound expression. The function must be one that can operate on the type of data that is returned by the named expression.

Example 1: Simple Advanced Expression

The following named expression, which identifies a text string, can be used as a prefix to the `AFTER_STR("<string>")` expression function, which works with text data:

```
HTTP.REQ.BODY(1000)
```

If the name of the expression is `top1000bytes`, you can use `top1000bytes.AFTER_STR("username")` instead of `HTTP.REQ.BODY(1000).AFTER_STR("username")`.

Example 2: Compound Advanced Expression

The name of the following compound named expression, which identifies a number, can be used in a comparison:

```
HTTP.REQ.HEADER("Header1").LENGTH + HTTP.REQ.HEADER("Header2").LENGTH
```

If the name of the compound expression is `headerlimit`, this enhancement allows you to use `headerlimit.GT(30)` instead of `(HTTP.REQ.HEADER("Header1").LENGTH + HTTP.REQ.HEADER("Header2").LENGTH) > 30`.

Additionally, you can use a named expression (either by itself or as a prefix to a function) to create the text expression for the replacement target in rewrite.

Support for Unsigned Long and Double Data Types

In NetScaler 9.3, simple expressions can return both double and unsigned long data types, thus allowing compound expressions that use arithmetic operators and logical operators to evaluate or return values of these data types. Additionally, you can use double and unsigned long values in policy expressions. For information about the functions that can work with the double and unsigned long data types, see the "Functions for Data Types in the Policy Infrastructure" section in the "Configuring Default Syntax Expressions: Getting Started" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Modifying (Corrupting) HTTP Headers

A new rewrite action, CORRUPT_HTTP_HEADER, replaces the name of the given HTTP header with a corrupted name so that it will not be recognized by the receiver.

Pattern Set Name Length

The name of a pattern set can now contain up to 127 characters.

Encryption and Decryption of Payloads

You can now configure the NetScaler appliance to encrypt and decrypt text and XML data in requests and responses. To encrypt and decrypt text, you use the ENCRYPT and DECRYPT functions, respectively. To encrypt and decrypt XML payloads, you use the XML_ENCRYPT() and XML_DECRYPT() functions, respectively. For more information about encrypting text, see the "Encrypting and Decrypting Text" section in the "Default Syntax Expressions: Evaluating Text" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673. For more information about encrypting XML payloads, see the "Encrypting and Decrypting XML Payloads" section in the "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Policy-Based Logging for Responder Policies

You can now log audit messages in a defined format when the rule in a responder policy evaluates to TRUE. To do this, you must configure an audit message action and associate it with a responder policy. The format of audit messages can be defined only with advanced expressions in an audit message. Audit message actions can be used to log messages at various log levels, either only in SYSLOG format or in both SYSLOG and newnslog formats. You can also define the SYSLOG messages to be logged. To log such SYSLOG messages, you must select **User Configurable Log Messages** (**System** > **Auditing** > **Change Global Auditing Settings**) in the NetScaler configuration utility or set the "UserDefinedLogging" SYSLOG parameter to YES in the NetScaler command line.

For example, you can configure a log action that defines the format of the log message and then associate the log action with a responder policy that can be bound globally or that can be bound to a load balancing or a content switching virtual server, as demonstrated in the following commands:

```
 Done
> set syslogparams -userDefinedAuditlog yes
 Done
> add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS Response Time to Servers:\" + sys
 Done
> add responder action redirect_url_2_act redirect "\"http://redirect_url.com\""
 Done
```

```
> add responder policy  redirect_policy_url_1  "http.req.url.eq(\"url_1.com\")" redirect_url_2_act RESET -log
 Done
> bind responder global redirect_policy_url_1 200 END -type REQ_DEFAULT
 Done
>
```

Support for the UTF-8 Character Set

The NetScaler policy infrastructure now supports the UTF-8 character set. For more information, see the “Specifying the Character Set in Expressions” section in the chapter, “Configuring Advanced Expressions: Getting Started,” in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Matching Text by Using a String Map

You can now use string maps to perform pattern matching in all NetScaler features that use the default policy syntax. A string map is a NetScaler entity that consists of key-value pairs. The keys and values are strings in either ASCII or UTF-8 format. String comparison uses two new functions, `MAP_STRING(<string_map_name>)` and `IS_STRINGMAP_KEY(<string_map_name>)`.

A policy configuration that uses string maps performs better than one that does string matching through policy expressions, and you need fewer policies to perform string matching with a large number of key-value pairs. String maps are also intuitive, simple to configure, and result in a smaller configuration.

For more information about string maps, see the "String Maps" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Rules for Names in Identifiers Used in Policies

The names of identifiers in the named expression, HTTP callout, pattern set, and rate limiting features must begin with an ASCII alphabet or an underscore (_). The remaining characters can be ASCII alphanumeric characters or underscores (_).

The names of these identifiers must not begin with the following reserved words:

- The words `ALT`, `TRUE`, and `FALSE` and the one-character identifiers `Q` and `S`.
- Words that indicate special syntax, which are `RE` (for regular expressions) and `XP` (for XPath expressions).
- Expression prefixes, which currently are the following:
  - CLIENT
  - EXTEND
  - HTTP

- SERVER
- SYS
- TARGET
- TEXT
- URL
- MYSQL

Additionally, the names of these identifiers cannot be the same as the names of enumeration constants used in the policy infrastructure. For example, the name of an identifier cannot be `IGNORECASE`, `YEAR`, or `LATIN2_CZECH_CS` (a MySQL character set).

Note: The NetScaler appliance performs a case-insensitive comparison of identifiers with these words and enumeration constants. For example, names of the identifiers cannot begin with `TRUE`, `True`, or `true`.

Stripping Characters From a String

You can use the `STRIP_CHARS(<string>)` operator to remove specific characters from the text that is returned by an advanced expression prefix. You can then use a text method on the resulting string. For example, you can compare the resulting string with the strings in a pattern set. You can also use the `STRIP_START_CHARS(<string>)` and `STRIP_END_CHARS(<string>)` functions to strip characters from the beginning and end of input strings, respectively. For more information about stripping characters from an input string, see the "Stripping Specific Characters from a String" section in the "Default Syntax Expressions: Evaluating Text" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Tool for Converting Classic Expressions to the Newer Default Expression Syntax

You can convert a classic expression to the default expression syntax by using the `nspepi` conversion tool. You can also use the tool to convert all the classic expressions in the NetScaler configuration to the default syntax (with the exception of NetScaler entities that currently support only classic expressions). For more information about converting classic expressions and NetScaler configuration files to the default syntax, see "Converting Classic Expressions to the Newer Default Expression Syntax" section in the "Introduction to Policies and Expressions" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Identifying a Response That is Associated with an HTTP Redirect

You can use the IS_REDIRECT function to determine whether a response is associated with a redirect. For information about the IS_REDIRECT function, see the "Expressions for HTTP Status Codes and Numeric HTTP Payload Data Other Than Dates" section in the "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data" chapter of the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Expressions for Generating the Day of the Week, as Strings, in Short and Long Formats

Two functions, WEEKDAY_STRING_SHORT and WEEKDAY_STRING, have been introduced for generating the day of the week as a string, in short and long formats, respectively. The strings that are returned are always in English. The prefix used with these functions must return the day of the week in integer format and the acceptable range for the value returned by the prefix is 0–6. Therefore, you can use any prefix that returns an integer in the acceptable range. An UNDEF condition is raised if the returned value is not in this range or if memory allocation fails.

Following are the descriptions of the functions:

Function	Description
<prefix>.WEEKDAY_STRING_SHORT	Returns the day of the week in short format. The short form is always 3 characters long with the first character in upper case and the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT returns Sun if the value returned by the WEEKDAY function is 0 and Sat if the value returned by the prefix is 6.
<prefix>.WEEKDAY_STRING	Returns the day of the week in long format. The first character in the long form is always in upper case and the remaining characters are in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING returns Sunday if the value returned by the WEEKDAY function is 0 and Saturday if the value returned by the prefix is 6.

Enhancements to Expressions That Process Binary Strings

The default policy infrastructure has had some changes and improvements regarding handling of binary strings. These are particularly useful for handling protocols that have binary values in them.

The two parameters of the GET_SIGNED16, GET_UNSIGNED16, and GET_SIGNED32 functions have been changed:

- The first parameter is now an offset, instead of an index into a list of identically sized items. This change enables the functions to handle items that are not aligned on the boundaries required by indexes.
  - For 16-bit values: instead of using `N`, use `2*N` for an aligned value.
  - For 32-bit values: instead of using `N`, use `4*N` for an aligned value.
- The second parameter now takes a mnemonic value instead of the nonintuitive `0` or `1`. The values are `LITTLE_ENDIAN` and `BIG_ENDIAN`.
  - What used to be `0` is now `LITTLE_ENDIAN`.
  - What used to be `1` is now `BIG_ENDIAN`.
- Expressions using these functions will need to be changed.

Example:

HTTP.REQ.BODY(100).GET_SIGNED16(7, BIG_ENDIAN)

Additionally, the following new functions have been introduced. The functions that produce binary strings, numbered 2 through 9 in the following list, are particularly useful in TCP rewrite, as replacement strings for binary data.

1. `<text>.GET_UNSIGNED32(n, endianness)`

   This function returns the 32-bit unsigned binary integer at offset `n`. If the offset would make part or all of the value outside of the current text, an `UNDEF` is raised. In the above expression:

   - `n` is the number of bytes from the current position to the first byte of the binary integer.
   - The “endianness” can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`.
   - The value returned is an unsigned long number. This is different from the other `GET` functions for binary strings, which return a number.

   Example:

   HTTP.REQ.BODY(100).GET_UNSIGNED32(30, LITTLE_ENDIAN)

2. `<number>.SIGNED8_STRING`

   This function produces an 8-bit signed binary string representing the number. If the value is out of range an `UNDEF` is raised. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STRING

3. `<number>.UNSIGNED8_STRING`

   This function produces an 8-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The value returned is of type text.

Example:

HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED8_STRING

4. `<number>.SIGNED16_STRING(endianness)`

   This function produces a 16-bit signed binary string representing the number. If the value is out of range, an `UNDEF` is raised. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)

5. `<number>.UNSIGNED16_STRING(endianness)`

   This function produces a 16-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LITTLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN

6. `<number>.SIGNED32_STRING(endianness)`

   This function produces a 32-bit signed binary string representing the number. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).AFTER_STR("delim").GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_

7. `<unsigned_long_number>.UNSIGNED8_STRING`

   This function produces an 8-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIGNED_LONG_AT.ADD(12).UNSIGNED8_STRING

8. `<unsigned_long_number>.UNSIGNED16_STRING(endianness)`

   This function produces a 16-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The "endianness" can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LITTLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UN

9. `<unsigned_long_number>.UNSIGNED32_STRING(endianness)`

   This function produces a 32-bit unsigned binary string representing the number. If the value is out of range, an `UNDEF` is raised. The “endianness” can be either `LITTLE_ENDIAN` or `BIG_ENDIAN`. The value returned is of type text.

   Example:

   HTTP.REQ.BODY(100).AFTER_STR("delim2").GET_UNSIGNED32(0, BIG_ENDIAN).ADD(2).UNSIGNED32_STRIN

Rewriting TCP Payloads

You can now treat the payload of a TCP packet as a raw stream of bytes and perform rewrite actions on the payload regardless of the protocol that the TCP connection is transmitting. You can evaluate TCP traffic that is associated with only those services that are of type `TCP` and `SSL_TCP`. You can configure policies that use the `CLIENT.TCP.PAYLOAD(<integer>)` expression prefix and the `SERVER.TCP.PAYLOAD(<integer>)` expression prefix to evaluate TCP and SSL_TCP traffic received from clients and servers, respectively. With these prefixes, you can use all types of existing string manipulation functions to identify the strings that you want to rewrite. To bring the policies into effect, you bind the TCP rewrite policies to load balancing virtual servers and content switching virtual servers of type `TCP` and `SSL_TCP`. You can also bind them to the following new bind points:

- OTHERTCP_REQ_DEFAULT
- OTHERTCP_REQ_OVERRIDE
- OTHERTCP_RES_DEFAULT
- OTHERTCP_RES_OVERRIDE

Note: The term "OTHERTCP" is used in the context of the NetScaler appliance to refer to all TCP or SSL_TCP requests and responses that you want to treat as a raw stream of bytes regardless of the protocol that the TCP packets encapsulate.

The order of evaluation of policies bound to these policy banks is the same as that for policies that evaluate HTTP traffic: TCP rewrite policies are evaluated from the most specific bind point to the most general bind point. That is, policies at the override bind points are evaluated first, followed by those at the virtual server bind points, and finally, those at the default bind points.

Note: A TCP rewrite policy does not evaluate connections that are active at the time at which it is bound to a bind point, but it evaluates the TCP connections that are initiated after binding is performed.

The following rewrite actions are supported in TCP rewrite policies.

- INSERT_AFTER
- INSERT_BEFORE

- REPLACE
- DELETE
- INSERT_AFTER_ALL
- INSERT_BEFORE_ALL
- REPLACE_ALL
- DELETE_ALL
- DROP
- RESET

Even though you cannot rewrite data other than that in TCP payloads, you can use all the expressions that are available for evaluating TCP traffic and information associated with lower network layers such as that in IP packets and Ethernet frames. Based on the information retrieved, you can perform an appropriate rewrite action, selected from the preceding list, on the TCP payload.

Additionally, the following two transform types (values for the `transform` parameter in the `add rewrite policyLabel` CLI command) have been introduced for rewrite policy labels:

- OTHERTCP_REQ
- OTHERTCP_RES

When you create a rewrite policy label, you specify `OTHERTCP_REQ` as the transform type to specify that the policies that are bound to the policy label are request-based policies. You can specify `OTHERTCP_RES` as the transform type to specify that the policies that are bound to the policy label are response-based policies.

Following are some limitations of the TCP rewrite feature:

- Named expressions and HTTP callouts are not supported in TCP rewrite policies.
- If the payload length received is less than the expected payload length, the NetScaler waits for the remaining bytes. Eventually, the request times out.

Example:

The following rewrite action `c_ip` identifies the first 150 bytes of the TCP payload and then inserts the string `CLIENT_IP` and the IP address of the client before the string `\r\n\r\n`.

add rewrite action c_ip INSERT_BEFORE "CLIENT.TCP.PAYLOAD(150).BEFORE_STR(\"\r\n\r\n\")" "\"CLIENT_IP\"

AppExpert

The following AppExpert enhancements are available in this release.

Deployment Files for AppExpert Applications

In earlier NetScaler releases, when you exported an AppExpert application, only a template file was created, in GZIP format. The template file contained only application-configuration information. You had to specify deployment-specific information, such as public endpoints and backend servers, in the **AppExpert Template Wizard** when importing the template file. With release 9.3, when you export an AppExpert application, a deployment file is created along with the application template file. Both files are created in XML format. The template file contains configuration-specific information and the deployment file contains deployment-specific information. In the **AppExpert Template Wizard**, you can specify use of the deployment file when you import the template file, or you can manually specify all the deployment information. If you specify the deployment file, the NetScaler appliance uses the deployment information in the deployment file. For more information, see the chapter, "AppExpert Applications and Templates," in the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Roll Back Support for AppExpert Application Import

In this release, if an error occurs during the import of an AppExpert application, the configuration changes that were made during the import process are automatically rolled back.

Creating Application Templates from Content Switching Virtual Servers

In earlier releases, you could export a content switching virtual server configuration to an application template only from the **Content Switching Visualizer**. Additionally, the template file was in GZIP format and included only configuration information. No deployment information was exported. In this release, you can export a content switching virtual server either from the **Content Switching Visualizer** or from the **Content Switching Virtual Servers** pane. A deployment file is created along with the template file. Both files are in XML format. For more information about exporting a content switching virtual server, template files, and deployment files, see the "Creating and Managing Template Files" chapter of the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Application Firewall

The following Application Firewall enhancements are available in this release.

Document Type Definitions (DTD) Support for XML Security Checks

Support has been added for document type definitions (DTDs) in the **Application Firewall Imports** pane, **XML Schema** tab. You can upload DTDs to the Application Firewall and they will be used for the relevant XML security checks.

Audit Server Configuration for Application Firewall

Support has been added to the NetScaler GUI for configuring the Application Firewall logs, in the **Auditing Policies** pane. In this pane, you can configure auditing to any server by adding that server in the **Servers** tab, and then adding an Audit policy that refers to that server. The process is the same as for AAA.

Learning Visualizer for Deployed Application Firewall Rules

The Application Firewall learning visualizer is now available for learned rules that you have already deployed, as well as rules that are pending review. For more information about how to use the learning visualizer, see the "Learning" section in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

SNMP Traps for Application Firewall Security Checks

SNMP traps have been added for each of the Application Firewall security checks. You configure the traps in the NetScaler configuration utility, under **System** > **SNMP** > **Traps**. For more information, see the *Citrix NetScaler SNMP OID Reference Guide* at http://support.citrix.com/article/CTX128676.

Application Firewall Custom Variables for HTML/XML Error Objects

You can now use five new variables to write troubleshooting information into an Application Firewall HTML or XML error object. These variables are:

- **${NS_TRANSACTION_ID}.** The transaction ID that the Application Firewall assigned to this transaction.
- **${NS_APPFW_SESSION_ID}.** The Application Firewall session ID.
- **${NS_APPFW_VIOLATION_CATEGORY}.** The specific Application Firewall security check or rule that was violated.
- **${NS_APPFW_VIOLATION_LOG}.** The detailed error message associated with the violation.

- **${COOKIE("<CookieName>")}.** The contents of the specified cookie. For <CookieName>, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.

To use these variables, you embed them in the HTML or XML of the error page object as if they were any ordinary text string. For more information, see the "Imports" chapter in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Cookie Encryption, Cookie Proxying, and Adding Flags to Cookies

The Application Firewall now supports encryption and proxying of cookies, and can be configured to add flags to cookies. For information about configuring this feature, see the "The Cookie Consistency Check" section in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Limiting the Number of Files Uploaded to a Web Site

By default, there is no limit on the number of files that can be uploaded to your protected Web site by using an HTML Web form or properly structured XML request. You can now limit the number of file uploads allowed on your protected Web sites. At the NetScaler command prompt, type:

set appfw profile -fileUploadMaxNum <integer>

For <integer>, substitute a number between 0 and 65535. If you set this value to 0, that disables all file uploads. If you set it to 65535, that allows unlimited uploads rather than stopping at 65535.

In the configuration utility, in the **Configure Application Firewall Profile** dialog box, **Settings** tab, you select the **Max File Uploads** check box, and enter a number in text box.

New Signatures Feature

All Application Firewall signatures, SQL injection and cross-site scripting patterns are now combined into a single XML-based signatures object file. You can download this file to your local computer and edit it in a text editor to add to or modify the lists of SQL injection keywords and special strings, SQL injection transformations, and cross-site scripting allowed and denied tags, attributes, and patterns. For more information, see the "Signatures" chapter in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Configuration Wizard Added to Application Firewall

The Application Firewall configuration wizard is now available to assist users in performing initial configuration of a new Application Firewall. You access the wizard on the main **Application Firewall** pane, in the details area, under **Getting Started**, by clicking **Application Firewall Wizard**.

Excluding XML Elements and Attributes from Security Checks

You can now exempt elements and attributes from the XML SQL Injection and Cross-Site scripting checks. To exempt an element from either check at the NetScaler command line, you issue one of the following commands:

bind appfw profile <name> -XMLSQLInjection <string> [-isRegex (REGEX | NOTREGEX)] [-location <location>

bind appfw profile <name> -XMLXSS <string> [-isRegex (REGEX | NOTREGEX)] [-location <location>]

For <name>, you substitute the name of the profile. For <string>, you substitute a literal string or a Posix-complaint regular expression that describes the element or elements to exempt. If you typed a regular expression, you must include " -isRegex REGEX". If you typed a string, this parameter is optional. For <location>, you type the location where this element is found in the XML document.

Inverse Regular Expressions

You can now easily create an inverse regular expression in the Application Firewall when you are adding a relaxation or rule to supported security checks. You do this by putting an exclamation point (!) at the beginning of the expression. You can use this new regular expression notation in Start URL, Form Field Consistency, Cookie Consistency, HTML SQL Injection, Cross-Site Request Forgery, or HTML Cross-Site Scripting check relaxations, or in Deny URL or Field Format check rules.

Note: You cannot use a solitary exclamation point to create an inverse expression. There must be a valid regular expression that follows the exclamation point, or it is treated as a text element and not as a logical inverse.

Adding XQuery Injection Patterns to XPath

The XPATH Injection settings can be used to detect XQuery violations, by adding the XQUERY injection pattern(s) that you want to block, and setting the injection type to XQUERY. In the configuration utility, you do this as follows:

1. In the navigation pane, expand **Application Firewall**, and select **Signatures**.
2. In the details pane, select the default Xpath Injection file, and then click **Add**.

3. In the **Add Signatures Object** dialog box, **Name** field, type a name for your new custom Xpath Injection file. You can also type a comment in the **Comment** field that describes the purpose of this custom file.

4. In the lower left-hand corner of the **Add Signatures Object** dialog box, click **Manage SQL/XSS patterns**.

5. In the **Manage SQL/XSS patterns** dialog box, **Filtered Results** window, select **Patterns**, and then in the button row beneath the **Filtered Results** window, click **Add**.

6. In the **Create Signatures Folder** dialog box, **Attributes** area, **delimiter** drop-down list, choose **Any**.

7. In the **type** text box, type **XQUERY**, and then click **OK**.

8. If your new folder is not selected, select it, and then click **Add**.

9. In the **Create Signatures Folder** dialog box, choose **keyword**, and then click **OK**.

10. Repeat steps 8 and 9, but choose **Special String** this time.

11. Add your XQUERY keywords and special strings by clicking the **Add** button beneath the details window below the **Filtered Results** window, filling out the **Create Signature Item** dialog box, and then clicking **OK**.

Web-Based GUI Editor for Imported Files

A GUI-based editor has been added for editing imported files in the **Application Firewall Import** pane. Users can now select a previously imported file, and then click **Open**, to open that file in a Web-based text editor, where they can make changes to the file. This allows minor modifications to be made without having to export the file to the local system for editing.

XML SOAP Array Rule Added to XDOS Check

The XML SOAP Array rule has been added to the XML denial-of-security (XDoS) check. This check protects SOAP processors on your XML server from attacks launched through SOAP Arrays that exceed either individual or total limits on size and dimensions. The check verifies the following attributes of SOAP arrays in requests:

- **XMLMaxSOAPArraySize.** Checks for SOAP arrays that exceed the configured maximum total SOAP Array size across the request.

  XMLMaxSOAPArrayRank. Checks for SOAP arrays that exceed the configured maximum individual SOAP Array dimensions, or rank.

You configure the XML SOAP Array rule under **Application Firewall** --> **Profiles**, by selecting a profile, and then clicking **Open**. In **the Configure Application Firewall Profiles** dialog box, you select **XML Denial of Service**, and then click **Open**. In the **Modify XML Denial of Service Check** dialog box, you select **Soap Array Check**, and enable or disable the check by using the appropriate buttons. For more information, see the *Citrix Application Firewall*

Guide at http://support.citrix.com/article/CTX128677, XML Security Checks chapter, XML Denial of Service Check section.

Application Firewall/XML Signature Updates

Users can update the signatures and patterns on their Application Firewall by importing an updated signatures object file under the same name as an existing signatures object. Signatures updates support the following:

- By default, when updating a signatures object file, the Application Firewall will display the local or network location from which a signatures object file was originally imported. The user can choose a different location, or accept the default.
- Signatures each have an identification number. By default, when updating, any signature in the update file that uses the same signature identification number as an existing signature replaces the existing signature. This means that any modifications that the user made to existing signatures by editing the existing file will be overwritten unless the user also modifies the identification number to a number that is not already in use. Any existing signatures that have unique IDs are preserved. Any signatures in the update file that have unique IDs are added to the signatures object file.
- Any configuration that the user performed in the configuration utility is preserved during an update.
- All in-memory profile configurations that use information in the signatures object are updated immediately, without restarting the NetScaler VPX process, the NetScaler appliance, or the Application Firewall appliance.
- The complete update process requires less than a minute.

For instructions on updating signatures, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677, Signatures chapter.

Importing Cenzic Scan Results as XML File

You can now import results from a Cenzic scan as a signatures object. To do so, you must save the results as an XSLT file, and then import them as a signatures object. During the import process, you select the “Convert External Format” check box and then converting the XSLT file during import. To import a Cenzic results file, and then browse to and select the XSLT file. The results are imported as a signatures object.

Compression

The following compression feature enhancements are available in this release.

Policy Manager for Compression

The NetScaler configuration utility now includes a policy manager for the HTTP Compression feature. By using the Compression Policy Manager, you can configure compression policies and/or bind them to various request-time and response-time bind points from a single dialog box.

To configure compression policy bindings by using the Compression Policy Manager

1. Click **HTTP Compression**, and then, in the details pane, click **Compression policy manager**.

2. In the **Compression Policy Manager** dialog box, click **Request** or **Response**, depending on whether you want the policy you are binding to be evaluated at request time or at response time.

   Click the name of the bind point to which you want to bind the policy. An example of a bind point is **Override Global**.

   The policies that have already been bound to the bind point appear in the dialog box.

4. To insert a policy, click **Insert Policy**, and then, in the drop-down list that appears in the **Policy Name** column, do one of the following:

   - Click **New Policy** to open the **Create Compression Policy** dialog box. In this dialog box, you can create a compression policy. After you create the policy in this dialog box, click **Create** to bind the policy to the bind point.

   - Click the name of an existing policy to bind it to the bind point.

5. To unbind a policy, click the name of the policy, and then click **Unbind Policy**.

6. To regenerate priorities, click **Regenerate Priorities**. The priority values are modified to begin at 100, with increments of 10, without affecting the order of evaluation.

7. To modify a policy that is already bound, click the name of the policy, and then click **Modify Policy**. If the policy is a built-in policy, the **View Compression Policy** dialog box displays the details of the policy. If the policy is a user-defined policy, the **Configure Compression Policy** dialog box appears and you can modify the policy.

8. After you have configured policies and policy bindings for the bind point, click **Apply Changes**.

9. Click **Close**.

Configuration Utility

The following configuration utility enhancements are available in this release.

EdgeSight Monitoring

The HTML Injection feature in NetScaler has been renamed to EdgeSight® Monitoring in the NetScaler Configuration Utility. You can now configure EdgeSight Monitoring for both load balancing and content switching virtual servers and configure the EdgeSight services. You can register the NetScaler appliance to an EdgeSight server and also access the EdgeSight server from the NetScaler.

Note: If you had configured the EdgeSight Monitoring feature on the NetScaler appliance before upgrading to release 9.3, you must configure the feature again on the virtual servers.

IP Address Type Identification in VLAN Configuration Dialog Boxes

A new column, Type, has been added to the table in the IPs section of the Create VLAN and Configure VLAN dialog boxes. This column displays the IP address type (such as mapped IP, virtual IP, and subnet IP) for each IP address in the IP Addresses column.

Default Policy Format for NetScaler Features

Starting with NetScaler release 9.3, the newer advanced policy syntax is referred to as the "default syntax." Additionally, the policy configuration dialog boxes for NetScaler features that support both the older classic policy syntax and the default syntax no longer include the Classic Syntax and Advanced Syntax option buttons. By default, the policy configuration dialog boxes accept policy expressions that use the default syntax. The option buttons have been replaced by a hyperlink that enables you to switch between the classic syntax and the default syntax.

Viewing SYSLOG Event IDs

You can now view the unique identification number of every NetScaler SYSLOG event in the Syslog Viewer dialog box.

IP Address Range Support

Now, as an alternative to creating IP addresses one at a time, you can specify a consecutive range of IP addresses.

Customized System User Prompt

You can also customize the NetScaler command-line prompt for a user. Prompts can be defined in a user’s configuration, in a user-group configuration, and in the global configuration. The prompt displayed for a given user is determined by the following order of precedence:

1. Display the prompt as defined in the user's configuration.
2. Display the prompt as defined in the group configuration for the user’s group.
3. Display the prompt as defined in the system global configuration.

UTF-8 Support in Regular Expression Editors

The regular expression editor in the configuration utility now supports the UTF-8 character set and parses and displays details of regular expressions that consist of UTF-8 characters.

Documentation

The following documentation enhancements are available in this release.

NetScaler Documentation on eDocs

The 9.3 documentation suite is now available on Citrix® eDocs, the documentation portal. PDF versions of the guides will continue to be available on http://support.citrix.com and Configuration Utility (the NetScaler® graphical user interface). The salient features of the NetScaler® documentation library on eDocs are as follows:

1. **Intuitive organization.** Information is presented in a flatter and more intuitive manner. Each main node represents a release, and the subnodes represent specific features or tasks. In addition, the first few nodes cover basic configuration tasks to help new users adopt the product quickly.
2. **Global search.** Users can now search across releases and features. Users can also create their own search scopes.
3. **PDF on-the-fly.** Users can custom-create their own PDF documents by selecting information from specific nodes in eDocs.

Enhanced Context-Sensitive Online Help

The context-sensitive online help has been enhanced to display links to blogs, videos, articles, and documentation pertaining to the dialog box or pane from which the online help is invoked. This is made possible by an intelligent content aggregator that fetches results from key Citrix sites such as Knowledge Center, Citrix TV, and eDocs. The new help system is compatible with most popular Web browsers, such as Internet Explorer, Chrome, Safari, and Firefox.

Hardware

The following hardware enhancement is available in this release.

Support for 1G SFP Hot Swap

This release introduces support for SFP hot swap on the Intel 1G (e1k) driver for both fiber and copper SFPs, on the following NetScaler hardware platforms:

- MPX 7500/9500
- MPX 9700/10500/12500/15500
- MPX 9700/10500/12500/15500 10G and 10G FIPS

High Availability

The following high availability enhancements are available in this release.

Synchronizing Configuration Files in High Availability Setup

In a high availability (HA) setup, the `sync HA files` command now supports the following two new synchronization modes:

`sync misc`. Synchronizes all licenses and the configuration (rc.conf) file.

`Sync_all_plus_misc`. Synchronizes all files, including licenses and rc.conf file.

Enhanced Force Failover Warning Message

In a high availability configuration, the warning message for the force failover command now also includes the information the warning is based on.

HTML Injection (EdgeSight Monitoring)

The following HTML Injection enhancement is available in this release.

Details of Virtual Server and Service

When you enable HTML injection, you can view the name, IP address, and port number of the virtual server that load balanced the request and of the physical service that sent the response to the request.

Integrated Cache

The following integrated cache enhancements are available in this release.

Integrated Cache Support for Single Byte-Range Requests

The integrated cache can now serve partial content in response to single byte-range requests.

Built-in Content Group, Pattern Set, and Policies for the Integrated Cache

Release 9.3 includes a built-in integrated caching configuration that you can use for caching content. The configuration consists of a content group called ctx_cg_poc, a pattern set called ctx_file_extensions, and a set of integrated cache policies. In the content group ctx_cg_poc, only objects that are 500 KB or smaller are cached. The content is cached for 86000 seconds, and the memory limit for the content group is 512 MB. The pattern set is an indexed array of common file extensions for file-type matching. For more information about the built-in configuration, see the "Built-in Content Group, Pattern Set, and Policies for the Integrated Cache" section in the "Integrated Caching" chapter of the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Monitoring Utility

The following monitoring utility enhancement is available in this release.

Viewing Dashboard Functionalities on the Monitoring Utility

You can now view the dashboard functionalities in HTML on the Monitoring interface. You can no longer view the Java Dashboard on nCore version.

The HTML Dashboard feature provides the following new functionalities:

- **Comparative charts**:
  - The two comparative charts on the left side of the **Monitoring** page help in monitoring a built-in set of counters for all the global and entity-based system statistics.
  - The charts are auto-updated and depict the variation of two or more counters over time. You can view the related historical data by clicking the **Reporting** link.
  - Maximizing the charts opens the charts in a new window. You can also add or delete counters from this view.
  - In addition to the built-in comparative charts, you can also load, save, or delete custom charts.
- **Graphical view**:
  - You can view the performance statistics data graphically by clicking **Graphical View**.

- You can hide or unhide the counter plot by clicking the legend name.
- By default, the graphical view for multiple-entity stat commands is displayed for the entity that has the maximum value for a predefined counter. You can, however, choose a different entity by clicking **Counters** and view it graphically.
- You can add or remove counters and other entities by clicking **Counters** on the bottom right corner of the page.
- Multiple counters for multiple entities with a maximum of 16 data series can be plotted.
- The graph also displays historical data for a maximum period of five minutes.

- **Virtual server view**: You can view the statistics of multiple load balancing virtual servers in consecutive rows, which have fixed headers, with all the statistic attributes lined up in columns. This view is also applicable to other entities, such as GSLB virtual servers, services, and interfaces.

Gaps between Java Dashboard and HTML Dashboard: The following features which were supported on Java Dashboard are currently not supported on HTML Dashboard:

- Viewing the per-core usage and per-interface throughput charts, which can be obtained by clicking the **CPU Usage** and **InUse Memory** meters
- Customizing the chart type
- Viewing charts for bound entities
- Viewing the graphs displayed in the graphical view in an independent window
- Changing the units for the counters on the Y axis from Rate to Total and the other way round

Domain Name System (DNS)

The following Domain Name System enhancements are available in this release.

DNS Security Extensions on the NetScaler

You can configure DNS Security Extensions (DNSSEC) on the Citrix® NetScaler® appliance. You can configure DNSSEC for zones for which the NetScaler appliance is authoritative. You can configure the NetScaler appliance as a DNS proxy server for signed zones hosted on a farm of backend name servers. If the NetScaler appliance is authoritative for a subset of the records belonging to a zone for which the appliance is configured as a DNS proxy server, you can include the subset of records in the DNSSEC implementation. For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Controlling the Number of Pipeline Requests on an Individual Client Connection

You can now control the number of pipelined DNS requests on a single client connection, which is identified by the <clientip:port>-<vserverip:port> tuple.

NetScaler VPX

The following NetScaler VPX enhancements are available in this release.

3 Gbps Throughput on the NetScaler Virtual Appliance (nCore VPX only)

The Citrix NetScaler VPX virtual appliance now supports a throughput of 3 gigabits per second (Gbps), which requires a 3 Gbps license on the virtual appliance.

VMTools Support (nCore VPX only)

VMTools provides the ability to perform graceful shutdown and reset of the virtual appliances by using the vSphere/vCenter clients. VMTools now reports the version of VMTools running, and also the state of each virtual appliance, to the ESX server. You can also use VMTools to directly enter the IP address, gateway, and subnet mask of each virtual appliance without starting the virtual appliance.

VLAN Tagging (nCore VPX only)

The NetScaler virtual appliance hosted on VMware ESX or ESXi platform now supports VLAN tagging.

Networking

The following networking enhancements are available in this release.

RNAT NAT IP Range

When creating a RNAT entry using the NetScaler command line, you can now specify a range of IP addresses for the **natip** parameter. RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified.

PBR with Multiple Next Hops

Instead of sending the selected packets to a next hop router, you can now configure the PBR to send them to a link load balancing (LLB) virtual server to which you have bound multiple next hops. So if one of the next hop fails, all the packets that matched against a PBR are routed to other bound next hops as determined by the LB method configured on the LLB virtual server.

Round Robin Method for Selecting Source IPs for IP Tunnels

Now you can set an IP tunnel global parameter that enables the NetScaler appliance to use a different source IP address for each new session through a particular IP tunnel, as determined by round robin selection of one of the SNIP addresses. This setting is ignored if a common global source IP address has been specified for all the IP tunnels. This setting does not apply to a tunnel for which a source IP address has been specified.

Simple ACL6

Now you can quickly create ACLs, called simple ACL6s that filter IPv6 packets on the basis of their source IP address and, optionally, their destination port and/or their protocol. Any packet that has the characteristics specified in the ACL is dropped.

Inbound Network Address Translation (INAT)

The following Inbound Network Address Translation (INAT) configurations are now supported:

- **IPv4-IPv6 Mapping:** A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance creates an IPv6 request packet with the IP address of the IPv6 server as the destination IP address.
- **IPv6-IPv4 Mapping:** A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance creates an IPv4 request packet with the IP address of the IPv4 server as the destination IP address.
- **IPv6-IPv6 Mapping:** A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.

Prefix-Based IPv6-IPv4 Translation

Now the NetScaler appliance can translate packets sent from private IPv6 servers into IPv4 packets, using an IPv6 prefix configured in the NetScaler appliance. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix.

The NetScaler appliance compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix. If there is a match, the NetScaler appliance generates an IPv4 packet and sets the destination IP address as the last 32 bits of the destination IP address of the matched IPv6 packet.

IPv6 packets addressed to this prefix have to be routed to the NetScaler so that the IPv6-IPv4 translation is done by the NetScaler.

Pinging Link Local Address from Any VLAN

An option that lets you specify a VLAN ID has been introduced in the `Ping6` command for pinging a link local address from the VLAN.

Host Name-Based SNMP Managers

While adding an SNMP manager, you can now specify an associated host name of the SNMP manager instead of its IP address. To do so, you need to add a DNS name server that resolves the host name of the SNMP manager to its IP address. You can add up to a maximum of five host name-based SNMP managers in addition to hundred IP address-based managers.

Time-out for Dynamic ARP Entries

You can now globally set an aging time (time-out value) for dynamically learned ARP entries. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time. You can specify an ARP time-out value from 1 through 1200 seconds.

Alias Name for a VLAN

An option for adding alias names for VLANs has been introduced, with a view to enhancing readability. However, you cannot perform VLAN operations by specifying alias names instead of VLAN IDs.

Option for Tagging or Untagging the NSVLAN

An option for tagging or untagging the NSVLAN (the VLAN to which NSIP address is bound) has been introduced. By default, the NSVLAN is tagged.

New vtysh Commands

The following vtysh commands have been introduced:

- **`show ns ha sync-status`:** Displays ZebOS config sync status.
- **`ns ha sync`:** Synchronizes ZebOS routing configuration between the primary and secondary nodes in a high availability configuration.

NITRO API

The following NITRO API enhancements are available in this release.

NITRO API Support for AppExpert Applications

NITRO API support has been added for exporting and importing of AppExpert applications.

Exporting an AppExpert application. When you export an AppExpert application, the following two XML files are automatically created:

- **Application template file**. Contains the AppExpert application's configuration information, such as application units, rules, and configured policies. The template file is stored in the /nsconfig/nstemplates/applications directory on the NetScaler appliance.
- **Deployment file**. Contains deployment-specific information of the AppExpert application, such as public endpoints, services, and configured variables. By default, the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory on the NetScaler appliance.

Importing an AppExpert application. You can import an AppExpert application template file, or a template file and its corresponding deployment file, to a NetScaler appliance. If you import both files, all of the application's configuration information is imported from the template file and all deployment-specific information is imported from the deployment file. Alternatively, if you import just the template file, you can set parameters to configure your own deployment settings (such as public endpoints (VIPs), services, and variable values).

NITRO API Support for Entity Templates

You can now import and export entity templates for a load balancing virtual server through the NITRO API version of the NetScaler appliance. The templates are saved as XML files.

Secure Sockets Layer (SSL)

The following Secure Sockets Layer enhancements are available in this release.

SHA-2 Signature Algorithm Support

The NetScaler appliance now supports client certificate authentication for certificates signed with the SHA-2 signature algorithm.

Small Records Processing

You can now insert the PUSH flag into decrypted, encrypted, or all records. If the PUSH flag is set to a value other than 0, the buffered records are forwarded to the server on the basis of the value of the PUSH flag.

The SSL engine has been enhanced to better handle small records received from a client or a server. This change optimizes the handling of multiple small SSL records in one TCP packet. Together with the option to set the PUSH flag after encryption or decryption as described above, the new feature helps prevent the generation of small packets on the server or the client.

Server Name Indication

If Server Name Indication (SNI) is enabled on the virtual server, all client requests in which the client initiates the handshake for one domain and sends an HTTP request for another domain are now dropped by the NetScaler appliance.

Support for IPv6 Addresses in Online Certificate Status Protocol

The NetScaler appliance now supports IPv6 addresses in Online Certificate Status protocol (OCSP) for certificate management.

Inserting Complete Client Certificate in Online Certificate Status Protocol

The NetScaler appliance now inserts the complete client certificate in its query for certificate status to the OCSP responder.

Advanced Encryption Standard New Instructions

Advanced Encryption Standard (AES) New Instructions (AES-NI) is now supported on the NetScaler VPX virtual appliance. AES-NI is supported on systems with Intel Westmere processors. On appliances with this processor, SSL bulk encryption performance for AES cipher will be significantly enhanced.

PUSH Flag-Based Encryption Trigger Mechanism

The encryption trigger mechanism that is based on the PUSH TCP flag now enables you to do the following:

- Merge consecutive packets in which the PUSH flag is set into a single SSL record or ignore the PUSH flag on packets.
- Perform timer-based encryption, in which the time-out value is set globally using the `set ssl parameter -pushEncTriggerTimeout <positive_integer>` command.

The options that are available for the PUSH encryption trigger enhancements are:

- **Always.** Any PUSH packet triggers encryption
- **Ignore.** Ignore PUSH packet for triggering encryption
- **Merge.** For consecutive sequence of PUSH packets, last PUSH packet triggers encryption
- **Timer.** PUSH packet triggering encryption delayed by timer period defined in '`set ssl parameter`'

The default value is "Always."

Importing an External Key as a FIPS Key (nCore MPX FIPS appliance only)

You can now import an external key as a FIPS key on the MPX 9700/10500/12500/15500 10G FIPS appliances. The key should be in PEM format and not be encrypted. When importing an external key on the MPX appliances, you do not need a wrap key, nor do you have to convert the key to PKCS8 format.

SSL Enhancements in the NetScaler Configuration Utility

Configuring the NetScaler for SSL Offloading has been made simpler with the following enhancements to the SSL landing page in the NetScaler configuration utility:

- Direct links to create a server certificate, root CA certificate, intermediate CA certificate, and client certificate.
- Access to the OpenSSL interface from the configuration utility.

In addition:

- When you remove a certificate-key pair, you are prompted to remove the certificate file from the appliance.
- To avoid errors in linking certificates, when you select a certificate and click Link, only certificates that can be linked to the selected certificate are listed.

Warm Restart Option On 10G FIPS Appliances (nCore only)

Warm restart is now supported on MPX 9700/10500/12500/15500 10G FIPS appliances.

System

The following system enhancements are available in this release.

Enhanced Application Visibility Using AppFlow (nCore and nCore VPX only)

The Citrix® NetScaler® appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called flow records, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

Rate Limiting Notification (nCore and nCore VPX only)

You can now configure a threshold to trigger an alarm to send SNMP traps when the throughput rate (in Mbps) reaches the specified threshold. You must configure the threshold to a value lesser than the maximum throughput of the NetScaler appliance. The maximum throughput is based on the license installed on the NetScaler appliance. The platform sends the platformRateLimitThresholdHigh trap when the throughput rate exceeds the high threshold. When the throughput rate returns to the normal limit, the platformRateLimitThresholdNormal trap is sent.

You can use the following command to set the snmp alarm and specify the maximum and the normal threshold limits:

set snmp alarm PF-RL-RATE-THRESHOLD [**-thresholdValue** <positive_integer> [**-normalValue** <positive_integer>]] [**-time** <secs>] [**-state** (**ENABLED** | **DISABLED**)] [**-severity** <severity>] [**-logging** (**ENABLED** | **DISABLED**)]

Optimizing the TCP Maximum Segment Size for a Virtual Server Configuration

In earlier releases, when a client connects to a virtual server on the NetScaler appliance, the appliance responds with either the default Maximum Segment Size (MSS) value of 1460 or the MSS advertised by the client. Now, you can specify the MSS that the appliance advertises to a client when the client initiates a connection to a virtual server on the appliance. You can configure the MSS for the virtual servers configured on the appliance in two ways:

- You can set the MSS for each virtual server to a value of your choice in a TCP profile.
- You can set the learnVsvrMSS global TCP parameter to ENABLED to enable MSS learning for all the virtual servers configured on the appliance.

For more information about optimizing the TCP maximum segment size for a virtual server configuration, see the "Advanced Configurations" chapter of the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Specifying a TCP Buffer Size Globally and for Virtual Servers and Services

Now, you can set the TCP buffer size both globally and for individual virtual servers and services through TCP profiles. The value that you set is the minimum value that is advertised by the NetScaler appliance and reserved when clients initiate a connection that is associated with an endpoint application function such as compression or SSL. If the TCP buffer size is set both at the global level and at the entity level (virtual server or service level), the buffer size that is specified for the virtual server takes precedence over the global value. For more information about specifying a TCP buffer size globally and for virtual servers and services, see the "Advanced Configurations" chapter of the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Memory Consumption Statistics for NetScaler Features

You can run the following command at the command prompt to view the NetScaler memory currently allocated for a particular feature, the percentage of memory allocated for the feature, and the total number of allocation failures for the feature:

```
stat ns memory
```

Statistics for Overall System Memory Consumption

You can view statistics for overall system memory consumption by running the following command at the NetScaler command prompt:

```
Stat system memory
```

The output of the command includes information about total available system memory, memory allocated in megabytes, percentage of memory allocated, in-use memory in megabytes, percentage of memory in use, and shared memory.

Filtering a NetScaler Trace to Capture Information About an Interface or a VLAN ID

You can now run a trace on the NetScaler appliance and filter packets associated with a given interface or VLAN ID. Only the packets that are filtered are captured in the trace files. To filter the trace for a given interface or VLAN ID, you must include the filter option in the nstrace command. The option takes a simple or compound expression. The expression must specify the interface or VLAN ID for which the trace must be filtered.

To capture the trace for a given interface on the appliance, at the NetScaler command prompt, type the following command:

nstrace -filter <expression>

The following example captures the trace at interface 1/1 by using a simple expression:

```
nstrace -filter "intf == 1/1"
```

The following example captures the trace at interface 1/1 and further filters the trace for packets associated with TCP destination port 80:

```
nstrace -filter "intf == 1/1 && destport == 80"
```

To capture the trace for a given VLAN ID, at the NetScaler command prompt, type the following command:

nstrace -filter "vlan <operator> <vlanid>**"**

The operator in the expression for filtering the trace on the basis of a VLAN ID can be equal to (==), not equal to (!=), less than (<), greater than (>), less than or equal to (<=), or greater than or equal to (>=).

The following example captures the trace for a VLAN with ID equal to 1:

```
nstrace -filter "vlan == 1"
```

Time Stamp of the Last Configuration Change

The “show nsconfig” command now displays the time stamp of the most recent change in the running configuration on the NetScaler.

View System Date in the NetScaler Command Line

The show ns config now displays the system date and time. The date and time of the NetScaler appliance is the date and time of the location in which the NetScaler is located. To change the system date and time, you must use the shell interface to the underlying FreeBSD OS.

Omitting Device-Specific Information in Output

An option `ignoredevspecific` has been introduced in the `diff ns config` command for not displaying device-specific information (such as IP address and host name) in the output.

Web Interface on NetScaler (nCore and nCore VPX only)

The Web Interface on NetScaler is based on Java Server Pages (JSP) technology and provides access to Citrix® XenApp™ and Citrix® XenDesktop® applications. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in.

The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance. The Web Interface sites provide user access to the XenApp and XenDesktop resources, which include applications, content, and desktops.

For information about the Web Interface feature, see the "Advanced Configuration" chapter of the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Traffic Management

The following traffic management enhancements are available in this release.

Load Balancing of Branch Repeaters for WAN Optimization (nCore and nCore VPX only)

Now, to provide high-scale WAN optimization, the branch repeaters deployed at data centers can be load balanced through NetScaler appliances. The bandwidth and number of concurrent sessions can be improved significantly.

DataStream™ (nCore and nCore VPX only)

The DataStream feature of NetScaler provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a decision based on SQL query parameters, such as user name and database name, command parameters. You can further configure monitors to track the state of database servers.

The advanced policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with `MYSQL.CLIENT` and `MYSQL.REQ`) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with `MYSQL.RES`) to evaluate server responses to user-configured health monitors. For more information about these expressions, see the "Evaluating Connections to Database Servers" chapter in the *Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: DataStream is supported only for MySQL databases. For information about the DataStream feature, see the "DataStream" chapter in the *Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

XenApp and XenDesktop Load Balancing Wizard Enhancements

Now, through the wizards for load balancing Citrix XenApp and Citrix XenDesktop, you can do the following:

- Choose to use the HTTP protocol for the backend communication even when the protocol for front-end communication is SSL.
- Specify any port number or use the port number specified on the virtual server.
- By default, the wizard selects the site path for the latest version of WI 5.x. You can select any path or specify a path of your choice.

No-Monitor Option for Services and Service Groups

If you do not want the NetScaler appliance to monitor a service or service group, you can disable monitoring by using the `healthMonitor` option for that service or service group. When the healthMonitor option is set to `NO`, the NetScaler appliance does not monitor the health of the service, and the service is always shown as `UP`.

Hash-Based Load Balancing Method

The hash-based load balancing algorithms have been enhanced to support pseudo-group persistency across multiple virtual servers. This pseudo-group persistency can be achieved by specifying a hash ID, which will be used for calculating the hash.

Support for Remote Desktop Protocol Load Balancing (nCore and nCore VPX only)

Starting this release, the NetScaler appliance supports Remote Desktop Protocol load balancing with session persistency. If the user reconnects with the session cookie, the NetScaler appliance connects the client to the server to which it was previously connected.

Policy-Based Routing Domains (nCore and nCore VPX only)

Now, enabling the L2Conn parameter for a load balancing virtual server allows multiple TCP and non-TCP connections with the same 4-tuple (<source IP>:<source port>::<destination IP>:< destination port>) to co-exist on the NetScaler appliance. The appliance uses both the 4-tuple and the Layer 2 parameters to identify TCP and non-TCP connections. The L2Conn parameters are the MAC address, VLAN ID, and channel ID, which are the Layer 2 parameters.

You can enable the L2Conn option in the following scenario:

1. Multiple VLANs are configured on the NetScaler appliance, and a firewall has been set up for each VLAN.

2. You want the traffic originating from the servers in one VLAN and bound for a virtual server in another VLAN to pass through the firewalls configured for both VLANs.

Note: In NetScaler 9.3 Classic, the l2Conn parameter, which specifies that Layer 2 connection parameters must also be used to uniquely identify a connection, is not supported for load balancing virtual servers even though it is listed in the CLI command synopses for load balancing virtual servers. The parameter is supported only for cache redirection virtual servers. In NetScaler 9.3 nCore, the parameter is supported for both load balancing virtual servers and cache redirection virtual servers. Therefore, when an nCore NetScaler appliance on which the l2Conn parameter is set for one or more load balancing virtual servers is downgraded to a Classic build or to an nCore build that does not support the l2Conn parameter, the load balancing configurations that use the l2Conn parameters are lost.

Graceful Shutdown of Backend Services

When you disable a service, if you cannot estimate the approximate amount of time it takes for all the connections of the service to complete their transactions, you can choose to shut down the service gracefully. In case of a graceful shutdown, the service is moved out of the service state only after all the clients connected to that service have either completed their transactions or are closed.

Additional Statistics for Load Balancing Virtual Servers

Now, in the statistics of virtual servers, you can see at the same time the number of hits per second for all the virtual servers configured on the NetScaler.

Persistence Time-out Option

The default time-out value for any persistence other than the cookie persistence is two minutes. If the time-out value is set to less than two minutes, the following error message appears:

ERROR: "Timeout value out of range; enter a value between 2 minutes and 1440 minutes"

Virtual Servers Bound to a Service Group

A new command show servicegroupbindings <serviceGroupName> displays the load balancing virtual servers that are bound to a service group.

Domain-Name-Based Service Groups

You can now bind domain-name-based service (DBS) members to a service group, in addition to IP-address-based members. If you bind the member on the basis of its domain name, you need not reconfigure the member on the NetScaler whenever the IP address of the member changes. The NetScaler automatically detects such changes.

N-tier Cache Redirection

You can deploy NetScaler appliances in two tiers (layers), with the appliances in the upper tier load balancing those in the lower tier, and the appliances in the lower tier load balancing the cache servers. The two-tier cache redirection helps in handling huge amounts (several Gbps) of traffic.

Sessionless Load Balancing in IP Mode

You can do sessionless load balancing in the IP-based forwarding mode as well as in the MAC-based forwarding mode. For sessionless load balancing in the IP mode, you need not configure the IP address of the virtual server on the physical servers.

Customization of the HTTPONLY Flag (Classic only)

You can now customize the addition of the “httponly” flag in persistence cookies. At the NetScaler command line, type:

set lb parameter [**-httpOnlyCookieFlag** (ENABLED|DISABLED)]

The flag is enabled by default.

Enhancement of XenDesktop Monitors

The XD-DDC monitor for the Dynamic Desktop Controller (DDC) servers of Citrix® XenDesktop™ includes more intelligent health checks. The DDC monitor can validate the login credentials.

The new CITRIX-WEB-INTERFACE monitor for the Web Interface component monitors a dynamic page in the specified site path. The dynamic page helps to check for critical failures in resource availability.

You can use the CITRIX-WI-EXTENDED monitor to verify the validity of the login credentials, correct configuration of the monitor (for example, the site path), and the connection with the IIS server.

Link Load Balancing

You can now configure multiple link load balancing (LLB) routes by specifying the same virtual server name as the gateway.

Rule-Based Persistence for ANY Type Virtual Servers

You can configure the load balancing virtual servers that have the service type as "ANY" with rule-based persistence. This can be used for load balancing the Branch Repeater appliances for WAN optimization and for any other physical servers.

NetScaler 9.3 Known Issues and Workarounds

The following known issues have been identified in this release.

Note: Unless stated otherwise, the known issues apply to Citrix® NetScaler® 9.3 Classic, NetScaler 9.3 nCore™, and NetScaler® 9.3 nCore™ VPX™. Workarounds are included where applicable.

Access Gateway Issues

Issue ID 81494

If users access a Distributed File Share on a computer running Windows Server 2008 64-bit, a blank folder appears in the directory path.

Issue ID 83492

When users log on using clientless access, a JavaScript error might appear when the logon page opens.

Issue ID 83819

If you configure a load balancing virtual server and the destination port is 21, when users log on with the Access Gateway Plug-in, logon is successful but data connections do not go through. When you configure a load balancing virtual server, do not use port 21.

Issue ID 84894

When users log off from the Access Gateway Plug-in and then clear the cache in Internet Explorer and Firefox, users might receive an error message that says "Error. Not a privileged user." Access Gateway records an HTTP/1.1 403 Access Forbidden error message in the logs.

Issue ID 84915

If users attempt to open and edit a Microsoft Office file from Outlook Web Access, users might receive an error and the file takes a long time to open. To allow users to edit files from Outlook Web Access, do the following:

1. Create a clientless access Outlook Web Access Profile and enable persistent cookies.
2. Bind the Outlook Web regular expression to this profile.
3. Bind the profile so that is assumes the highest priority.

Issue ID 85861

If you enable **ICA Proxy** on Access Gateway, when users log on and attempt to open a virtual application, the connection times out and terminates.

Issue ID 85906

When users log on with an earlier version of the Access Gateway Plug-in, users do not receive the upgrade prompt and the user device receives a session ID. However, the session is not established and the Web browser trying to load the file services.html and upgrading the plug-in both fail.

Issue ID 86022

If you configure the user device to enable users to log on only using the Access Gateway Plug-in and then change the plug-in Web address to an unresolvable address, when users try to log on through the logon dialog box, an authentication error appears. Then, if users try to log on using the plug-in, the logon dialog box does not appear and users cannot change the Web address. Users should exit and then restart the plug-in to subsequently change the Web address.

Issue ID 86122

If you disable transparent interception and set the force time-out setting, when users log on with the Access Gateway Plug-in for Java, when the time-out period expires, a session time-out message appears on the user device, however the session is not terminated on Access Gateway.

Issue ID 86123

If users log on with clientless access in the Firefox Web browser, when users click a link for a virtual application, the tab closes and the application does not start. If users right-click the virtual application and attempt to open it in a new window, the Web Interface appears and users receive the warning "Published resource shortcuts are currently disabled." Users can open the virtual application in Internet Explorer.

Issue ID 86323

If you configure single sign-on with Windows and configure the user name with special characters, when users log on to Windows 7 Professional, single sign-on fails. Users receive the error message "Invalid username or password. Please try again." This issue does not occur if users log on to Windows XP.

Issue ID 86470 and 86787

When users log on with the Access Gateway Plug-in for Windows using Internet Explorer 9, a delay may occur in establishing the connection. The Access Interface, or a custom home page, might take a long time to appear when users log on using Internet Explorer 9.

Issue ID 86471

When users log on with the Access Gateway Plug-in by using a Web browser, users might see a delay during logon.

Issue ID 86722

When users log on with clientless access using Internet Explorer 9 and connect to SharePoint 2007, some images might not appear correctly.

Application Firewall Issues

Issue ID 77089

In certain cases, transformation does not work correctly for allowed/denied XSS patterns defined in custom settings file.

Issue ID 81616

Attempts to upload a 10 MB or larger file may fail when Cross-Site Scripting (XSS) and SQL Injection checks are enabled.

Issue ID 83088

Sig Rule 16218 blocks any request with Content-length Header and should only be enabled by the user after careful consideration since it will block all POST requests.

Issue ID 83089

The users can look at the default signature rules in GUI, but it will be useful to have a comprehensive list of all the rules accessible in documentation or white papers for users to review.

Issue ID 81650

Certain XML schemas may be imported successfully, but when the user attempts to bind that XML schema to an Application Firewall profile, the binding fails.

Workaround: Avoid using XML schemas that contain DOCTYPE definitions with references to a DTD.

Issue ID 83707

The Application Firewall does not import XML schema or WSDL files that contain non-ASCII characters correctly. Attempting to import such files results in a partial import.

Workaround: Convert XML schema or WSDL files to ASCII before importing them.

Issue ID 83754

The Application Firewall may fail to detect a signature violation in a request that contains trailing spaces after the PCRE match expression sent in an HTTP header.

Issue ID 86782

Importing Application Firewall objects, such as signatures and HTML error pages may fail on the first attempt if the import command contains a very long source URL. The command will succeed when executed for the second time.

Issue ID 82058

The "unique" element in the XML schema is currently not supported.

Issue ID 82059

The "redefine" element in the XML schema is currently not supported.

Issue ID 82069

When the Application Firewall validates XML messages, it does not validate the contents of elements that are defined as type "any" in the applicable XML schema. Specifically, it treats these elements as if the processContent attribute was set to "skip".

Workaround: Replace the "any" type definitions in the XML schemas with definitions of the actual elements that occur in the XML message. (The "any" type is rarely used.)

Command-Line Interface Issues

Issue ID 77865

In certain rare cases, connection to the NetScaler over Secure FTP (SFTP) by using superuser (administrator) credentials other than nsroot fails.

Configuration Utility Issues

Issue ID 87595

When upgrading the NetScaler appliance from 9.2 e release to 9.3 release using the **Upgrade wizard** in the configuration utility, if you keep the wizard idle for 5 minutes then the **Yes** and **No** radio buttons will disappear.

Issue ID 86201

If you use Internet Explorer 9 to log on to the NetScaler configuration utility, the following page is displayed: res://ieframe.dll/acr_error.htm.

The page displays the following message: "A problem with this Web page caused Internet Explorer to close and reopen the tab."

Workaround: Enable Compatibility View in the browser and then log on to the NetScaler configuration utility.

DataStream Issues

Issue ID 82084

On receiving the MySQL nit_db command from the client, the virtual server of type MySQL links the client connection permanently with a server-side connection and no further requests on that client connection are multiplexed.

Issue ID 82872

If the connection to the database server is a transaction or if special queries are sent to the database server, the maximum request per connection setting is overridden and all the queries are sent using the same connection.

Integrated Cache Issues

Issue ID 81159

When the NetScaler appliance receives a single byte-range request, if the starting position of the range is beyond 9 megabytes, the appliance sends the client a full response with a status code of `200 OK` instead of a partial response.

Load Balancing Issues

Issue ID 80170

The syntax of the `unset servicegroup` command has been changed to allow unsetting of the parameters of the service group members. This can cause XML API incompatibility with respect to the `unset servicegroup` command.

Issue ID 82929

If a TCP monitor is configured for a MySQL service, attempts by the NetScaler appliance to establish a connection with the MySQL server are treated as failed logon attempts. After some time, the MySQL server blocks the NetScaler mapped IP address (MIP) from making new MySQL connections and returns the following error message: "MYSQL Server Greeting Error 1129 Host <hostname> is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'"

Workaround: Bind a MYSQL_ECV monitor to the MySQL service.

Issue ID 82996

The MySQL monitor shows the state of the service as UP even if a mapped IP address (MIP) or subnet IP address (SNIP) is not configured on the NetScaler.

Issue ID 83862 (nCore and nCore VPX only)

In this release, IPv6 addresses are not supported in the DataStream feature.

Issue ID 83924 (nCore and nCore VPX only)

When the MySQL server responds with an error to a syntactically or semantically incorrect SET command that is sent by the client, the NetScaler appliance resets the

client connection.

Issue ID 86096

When you configure the WI-EXTENDED monitor, when specifying the site path, do not enter a slash (/) at the end of the path as the software internally adds a slash at the end of the path. For example, note the following command:

add monitor wi CITRIX-WI-EXTENDED -**sitepath** "/Citrix/DesktopWeb" **-username** aaa **-password** bbb **-domain** ccc

Issue ID 87407 (nCore and nCore VPX only)

When an RDP service is configured, the NetScaler appliance automatically maintains persistence through session cookies using Session Directory. You need not explicitly configure persistency on NetScaler. In the next releases IP address based persistency will be supported.

In some situations, (where multiple persons use same user-login credentials) session cookie persistence may not be helpful and IP-based persistence methods will be necessary. In some other situations, load balancing of the RDP services without persistence may be necessary. That is, each new connection to an RDP virtual server needs to be load balanced irrespective of a user's disconnected session existing on a terminal server.

Issue ID 87201 (Classic only)

On a load balancing virtual server for TCP services on which stateful connection failover is enabled, an established connection may be broken if a failover occurs more than once while a large amount of data is being transferred.

Issue ID 87242 (Classic only)

On a load balancing virtual server for FTP services on which connection failover is enabled and USIP mode is enabled for the service, if a failover occurs and the client sends a new FTP command before the CM connections are re-established, the established FTP control connection breaks.

NITRO API Issues

Issue ID 83274

When you use the NITRO API to run the `unset` commands for a service or service group, the NetScaler log file (ns.log) shows two `unset` commands and one associated `set` command instead of one `unset` command and one associated `set` command.

Reporting Issues

Issue ID 85025

The Reporting tool chart does not properly plot counters per packet engine.

SSL Issues

Issue ID 74279

The cipher TLS1-EXP1024-DES-CBC-SHA is not supported by the NetScaler appliance.

Issue ID 81825

If you import an external key as a FIPS key, you cannot create a certificate signing request with this key.

Issue ID 82908

In certain rare cases, when the NetScaler appliance is subject to conditions of heavy SSL-related traffic, CLI commands fail and report a configuration inconsistency error.

Workaround: Check for configuration inconsistency by using the "show configstatus" command and reconfigure the appliance under low traffic conditions or during a maintenance period. If this does not resolve the issue, restart the appliance.

Issue ID 80830 (nCore only)

When you attempt to delete an SSL certificate key-pair object that is referenced by a Certificate Revocation List (CRL), the message, " ERROR: Configuration possibly inconsistent. Please check with the "show configstatus" command or reboot," is displayed. This message is not the intended message. However, on the subsequent attempt to delete the certificate key-pair object, the correct message, " ERROR: Certificate is referenced by a CRL, OCSP responder, virtual server, service, or another certificate," is displayed.

MPX 9700/10500/12500/15500 10G FIPS Appliances Issues

Issue ID 81850 (nCore only)

You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 10G FIPS appliance.

Workaround: First, decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

openssl rsa -in <EncryptedKey.key> > **DecryptedKey.out**

Issue ID 85522 (nCore only)

Signed OCSP requests fail if the requests are encrypted by using an imported external FIPS key. This is a known bug in the crypto card's firmware, which returns "0" for the modulus of an imported external FIPS key.

System Issues

Issue ID 84282

If the global setting for the maximum segment size (mss) to use for TCP connections is less than 1220, the NetScaler appliance causes excessive delay to save the configuration.

Issue ID 65259

As a result of the Apache2 upgrade, customers need to remove any `/nsconfig/httpd.conf` file they have used to customize their httpd runtime environment. The older `httpd.conf` files are not forward-compatible with Apache2.

NetScaler VPX Issues

Issue ID 87605

For an instance of NetScaler VPX running on VMware ESX server, you need to enable promiscuous mode on the corresponding virtual switch (vswitch) port to configure the following on VPX:

- Virtual MAC address (VMAC)
- Link Aggregation (LA)
- LA with Link Aggregation Control Protocol (LACP)

Issue ID 87858

You can use the 9.3 xva images for XenServer to create a new VPX instance on XenServer version 5.6 and 5.6 FP1. You can also upgrade an existing VPX instance running on XenServer version 5.0 or version 5.5 to release 9.3. However, on appliances running XenServer version 5.0 or version 5.5, you cannot create a new VPX instance from the 9.3 xva image.

Getting Started with Citrix NetScaler

Intended for system and network administrators who install and configure complex networking equipment, this section of the library describes initial set-up and basic configuration of the NetScaler, including the following topics.

Understanding the NetScaler	What a NetScaler is and where it fits in a network, with descriptions of entities used in typical configurations and the order in which data is processed by the various features.
Introduction to the NetScaler Product Line	Brief introduction to the software and hardware platforms. Also covers the feature groups.
Installing the NetScaler Hardware	Tasks for unpacking and installing the hardware, including rack mounting, connecting transceivers, connecting the console cable, connecting to a power source, and connecting to a network.
Accessing a NetScaler	Descriptions of the CLI and GUI access mechanisms that you can use to configure and monitor a NetScaler.
Configuring a NetScaler for the First Time	Procedures for configuring a NetScaler for the first time.
Understanding Common Network Topologies	Describes the four common deployment topologies: Two-Arm Multiple Subnet, Two-Arm Transparent, One-Arm Single Subnet, and One-Arm Multiple Subnet. Includes topology diagrams, sample values, and references.
Configuring System Management Settings	Procedures for configuring basic system management settings, such as VLANs, SNMP, and DNS.
Load Balancing Traffic	Basic introduction to the load balancing feature. Includes procedures for configuring a basic load balancing setup to deliver a Web application, and procedures for configuring persistence, URL redirection, and backup vservers.
Accelerating Load Balanced Traffic by Using Compression	Basic introduction to the compression feature. Includes procedures for configuring a NetScaler to compress application traffic.

Securing Load Balanced Traffic by Using SSL	Basic introduction to the SSL offload feature. Includes procedures for configuring a NetScaler to secure application traffic by using SSL.
Features at a Glance	Brief description of all the features, with links to documentation for the features.

Understanding the NetScaler

The Citrix® NetScaler® product is an application switch that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4-L7) network traffic for web applications. For example, a NetScaler makes load balancing decisions on individual HTTP requests rather than on the basis of long-lived TCP connections, so that the failure or slowdown of a server is managed much more quickly and with less disruption to clients. The NetScaler feature set can be broadly categorized as consisting of switching features, security and protection features, and server-farm optimization features.

Understanding the NetScaler

The Citrix® NetScaler® product is an application switch that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4-L7) network traffic for web applications. For example, a NetScaler makes load balancing decisions on individual HTTP requests rather than on the basis of long-lived TCP connections, so that the failure or slowdown of a server is managed much more quickly and with less disruption to clients. The NetScaler feature set can be broadly categorized as consisting of switching features, security and protection features, and server-farm optimization features.

Switching Features

When deployed in front of application servers, a NetScaler ensures optimal distribution of traffic by the way in which it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP or TCP request, and on the basis of L4-L7 header information such as URL, application data type, or cookie. Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

Security and Protection Features

NetScaler security and protection features protect web applications from application-layer attacks. A NetScaler allows legitimate client requests and can block malicious requests. It provides built-in defenses against denial-of-service (DoS) attacks and supports features that protect applications against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from application-layer attacks, including buffer overflow exploits, SQL injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

Optimization Features

Optimization features offload resource-intensive operations such as Secure Sockets Layer (SSL) processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. This improves the performance of the servers in the server farm and therefore speeds up applications. A NetScaler supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

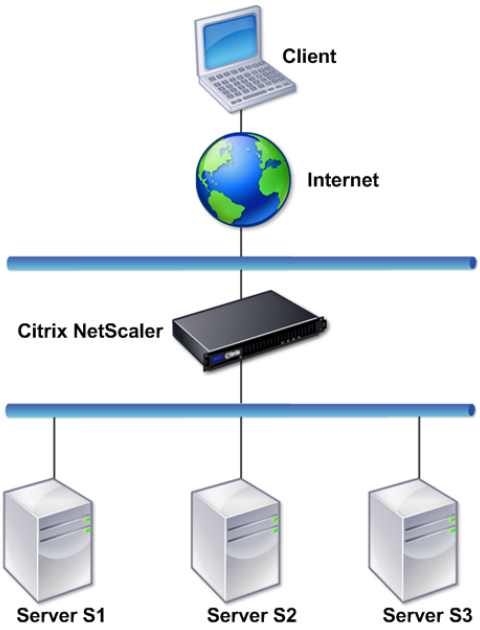
Where Does a NetScaler Fit in the Network?

A NetScaler resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers (vservers) configured on the NetScaler provide connection points that clients use to access the applications behind the NetScaler. In this case, the NetScaler owns public IP addresses that are associated with its vservers, while the real servers are isolated in a private network. It is also possible to operate the NetScaler in a transparent mode as an L2 bridge or L3 router, or even to combine aspects of these and other modes.

Physical Deployment Modes

A NetScaler logically residing between clients and servers can be deployed in either of two physical modes: inline and one-arm. In inline mode, multiple network interfaces are connected to different Ethernet segments, and the NetScaler is placed between the clients and the servers. The NetScaler has a separate network interface to each client network and a separate network interface to each server network. The NetScaler and the servers can exist on different subnets in this configuration. It is possible for the servers to be in a public network and the clients to directly access the servers through the NetScaler, with the NetScaler transparently applying the L4-L7 features. Usually, vservers (described later) are configured to provide an abstraction of the real servers. The following figure shows a typical inline deployment.

Figure 1. Inline Deployment



In one-arm mode, only one network interface of the NetScaler is connected to an Ethernet segment. The NetScaler in this case does not isolate the client and server sides of the network, but provides access to applications through configured vservers. One-arm mode can simplify network changes needed for NetScaler installation in some environments.

For examples of inline (two-arm) and one-arm deployment, see Understanding Common Network Topologies.

Citrix NetScaler as an L2 Device

A NetScaler functioning as an L2 device is said to operate in L2 mode. In L2 mode, the NetScaler forwards packets between network interfaces when all of the following conditions are met:

- The packets are destined to another device's media access control (MAC) address.
- The destination MAC address is on a different network interface.
- The network interface is a member of the same virtual LAN (VLAN).

By default, all network interfaces are members of a pre-defined VLAN, VLAN 1. Address Resolution Protocol (ARP) requests and responses are forwarded to all network interfaces that are members of the same VLAN. To avoid bridging loops, L2 mode must be disabled if another L2 device is working in parallel with the NetScaler.

For information about how the L2 and L3 modes interact, see Configuring Modes of Packet Forwarding.

For information about configuring L2 mode, see Enabling and Disabling Layer 2 Mode.

Citrix NetScaler as a Packet Forwarding Device

A NetScaler can function as a packet forwarding device, and this mode of operation is called L3 mode. With L3 mode enabled, the NetScaler forwards any received unicast packets that are destined for an IP address that it does not have internally configured, if there is a route to the destination. A NetScaler can also route packets between VLANs.

In both modes of operation, L2 and L3, a NetScaler generally drops packets that are in:

- Multicast frames
- Unknown protocol frames destined for a NetScaler's MAC address (non-IP and non-ARP)
- Spanning Tree protocol (unless BridgeBPDUs is ON)

For information about how the L2 and L3 modes interact, see Configuring Modes of Packet Forwarding.

For information about configuring the L3 mode, see Enabling and Disabling Layer 3 Mode.

How a NetScaler Communicates with Clients and Servers

A NetScaler is usually deployed in front of a server farm and functions as a transparent TCP proxy between clients and servers, without requiring any client-side configuration. This basic mode of operation is called Request Switching technology and is the core of NetScaler functionality. Request Switching enables a NetScaler to multiplex and offload the TCP connections, maintain persistent connections, and manage traffic at the request (application layer) level. This is possible because the NetScaler can separate the HTTP request from the TCP connection on which the request is delivered.

Depending on the configuration, a NetScaler may process the traffic before forwarding the request to a server. For example, if the client attempts to access a secure application on the server, the NetScaler might perform the necessary SSL processing before sending traffic to the server.

To facilitate efficient and secure access to server resources, a NetScaler uses a set of IP addresses collectively known as NetScaler-owned IP addresses. To manage your network traffic, you assign NetScaler-owned IP addresses to virtual entities that become the building blocks of your configuration. For example, to configure load balancing, you create virtual servers (vservers) to receive client requests and distribute them to services, which are entities representing the applications on your servers.

Understanding NetScaler-Owned IP Addresses

To function as a proxy, a NetScaler uses a variety of IP addresses. The key NetScaler-owned IP addresses are:

NetScaler IP address (NSIP)

The NSIP is the IP address for management and general system access to the NetScaler itself, and for HA communication.

Mapped IP address (MIP)

A MIP is used for server-side connections. It is not the IP address of the NetScaler. In most cases, when the NetScaler receives a packet, it replaces the source IP address with a MIP before sending the packet to the server. With the servers abstracted from the clients, the NetScaler manages connections more efficiently.

Virtual server IP address (VIP)

A VIP is the IP address associated with a vserver. It is the public IP address to which clients connect. A NetScaler managing a wide range of traffic may have many VIPs configured.

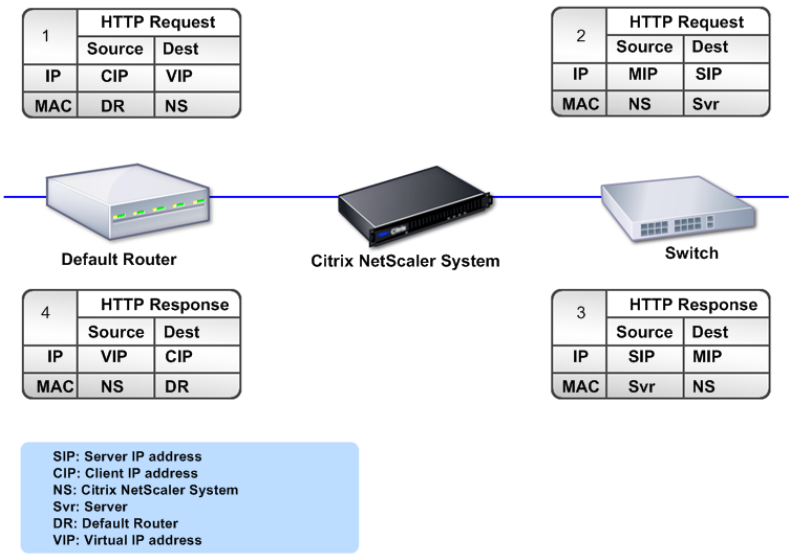
Subnet IP address (SNIP)

When the NetScaler is attached to multiple subnets, SNIPs can be configured for use as MIPs providing access to those subnets. SNIPs may be bound to specific VLANs and interfaces.

How Traffic Flows Are Managed

Because a NetScaler functions as a TCP proxy, it translates IP addresses before sending packets to a server. When you configure a vserver, clients connect to a VIP on the NetScaler instead of directly connecting to a server. Based on the settings on the vserver, the NetScaler selects an appropriate server and sends the client's request to that server. By default, the NetScaler uses a MIP or SNIP to establish connections with the server, as shown in the following figure.

Figure 1. Vserver-Based Connections



Note: You can use SNIP instead of MIP in the preceding figure.

In the absence of a vserver, when a NetScaler receives a request, it transparently forwards the request to the server. This is called the transparent mode of operation. When operating in transparent mode, a NetScaler translates the source IP addresses of incoming client requests to the MIP or SNIP but does not change the destination IP address. For this mode to work, L2 or L3 mode needs to be configured appropriately.

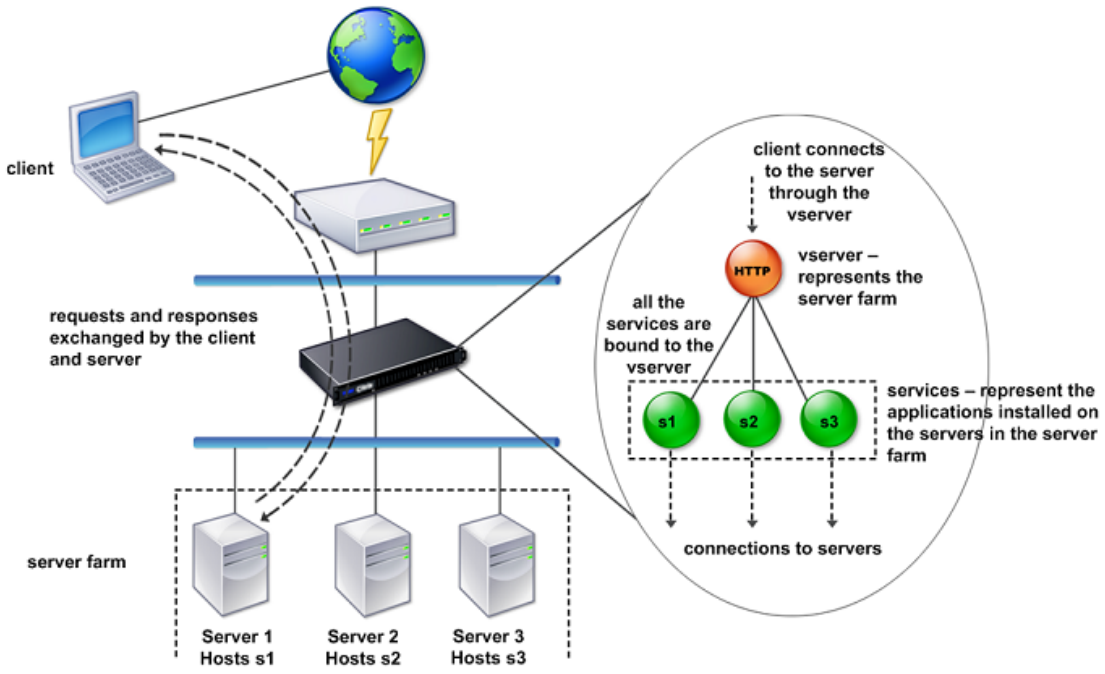
For cases in which the servers need the actual client IP address, the NetScaler can be configured to modify the HTTP header by inserting the client IP address as an additional field, or configured to use the client IP address instead of the MIP or SNIP for connections to the servers.

Traffic Management Building Blocks

The configuration of a NetScaler is typically built up with a series of virtual entities that serve as building blocks for traffic management. The building block approach helps separate traffic flows. Virtual entities are abstractions, typically representing IP addresses, ports, and protocol handlers for processing traffic. Clients access applications and resources through these virtual entities. The most commonly used entities are vservers and services. Vservers represent groups of servers in a server farm or remote network, and services represent specific applications on each server.

Most features and traffic settings are enabled through virtual entities. For example, you can configure a NetScaler to compress all server responses to a client that is connected to the server farm through a particular vserver. To configure the NetScaler for a particular environment, you need to identify the appropriate features and then choose the right mix of virtual entities to deliver them. Most features are delivered through a cascade of virtual entities that are bound to each other. In this case, the virtual entities are like blocks being assembled into the final structure of a delivered application. You can add, remove, modify, bind, enable, and disable the virtual entities to configure the features. The following figure shows the concepts covered in this section.

Figure 1. How Traffic Management Building Blocks Work

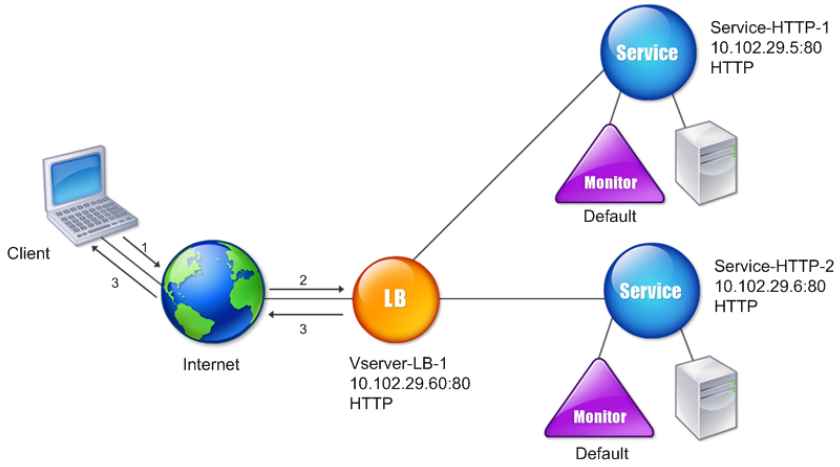


A Simple Load Balancing Configuration

In the example shown in the following figure, the NetScaler is configured to function as a load balancer. For this configuration, you need to configure virtual entities specific to load balancing and bind them in a specific order. As a load balancer, a NetScaler distributes client requests across several servers and thus optimizes the utilization of resources.

The basic building blocks of a typical load balancing configuration are services and load balancing vservers. The services represent the applications on the servers. The vservers abstract the servers by providing a single IP address to which the clients connect. To ensure that client requests are sent to a server, you need to bind each service to a vserver. That is, you must create services for every server and bind the services to a vserver. Clients use the VIP to connect to a NetScaler. When the NetScaler receives client requests on the VIP, it sends them to a server determined by the load balancing algorithm. Load balancing uses a virtual entity called a monitor to track whether a specific configured service (server plus application) is available to receive requests.

Figure 1. Load Balancing Virtual Server, Services, and Monitors



In addition to configuring the load balancing algorithm, you can configure several parameters that affect the behavior and performance of the load balancing configuration. For example, you can configure the vserver to maintain persistence based on source IP address. The NetScaler then directs all requests from any specific IP address to the same server.

Understanding Virtual Servers

A vserver is a named NetScaler entity that external clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP address (VIP), port, and protocol. The name of the vserver is only of local significance and is designed to make the vserver easier to identify. When a client attempts to access applications on a server, it sends a request to the VIP instead of the IP address of the physical server. When the NetScaler receives a request on the VIP, it terminates the connection at the vserver and uses its own connection with the server on behalf of the client. The port and protocol settings of the vserver determine the applications that the vserver represents. For example, a web server can be represented by a vserver and a service whose port and protocol are set to 80 and HTTP, respectively. Multiple vservers can use the same VIP but different protocols and ports.

Vservers are points for delivering features. Most features, like compression, caching, and SSL offload, are normally enabled on a vserver. When the NetScaler receives a request on a VIP, it chooses the appropriate vserver by the port on which the request was received and its protocol. The NetScaler then processes the request as appropriate for the features configured on the vserver.

In most cases, vservers work in tandem with services. You can bind multiple services to a vserver. These services represent the applications running on physical servers in a server farm. After the NetScaler processes requests received on a VIP, it forwards them to the servers as determined by the load balancing algorithm configured on the vserver. The following figure shows these concepts.

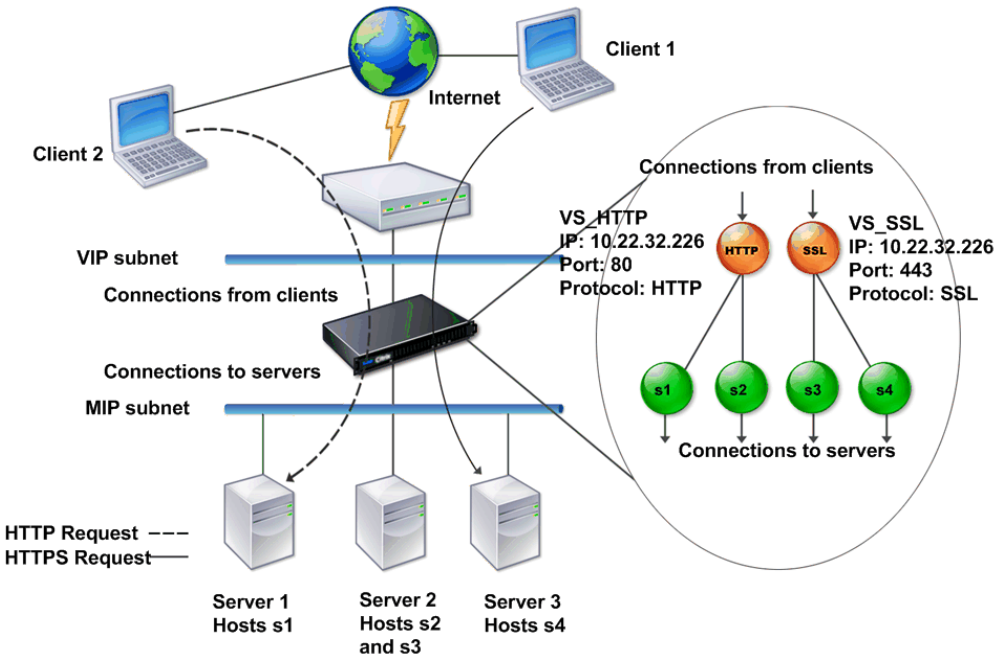


Figure 1. Multiple Virtual Servers on a Single VIP

The preceding figure shows a configuration consisting of two vservers with a common VIP but different ports and protocols. Each of these vservers has two services bound to it. The services s1 and s2 are bound to VS_HTTP and represent the HTTP applications on Server 1 and Server 2. The services s3 and s4 are bound to VS_SSL and represent the SSL applications on Server 2 and Server 3 (Server 2 provides both HTTP and SSL applications). When the NetScaler receives an HTTP request on the VIP, it processes the request based on the settings of VS_HTTP and sends it to either Server 1 or Server 2. Similarly, when the NetScaler receives an HTTPS request on the VIP, it processes it based on the settings of VS_SSL and it sends it to either Server 2 or Server 3.

Vservers are not always represented by specific IP address, port numbers, or protocols. They can be represented by wildcards, in which case they are known as wildcard vservers. For example, when you configure a vserver with a wildcard instead of a VIP, but with a specific port number, the NetScaler intercepts and processes all traffic conforming to that protocol and destined for the predefined port. For vservers with wildcards instead of VIPs and port numbers, the NetScaler intercepts and processes all traffic conforming to the protocol.

Vservers can be grouped into the following categories:

Load balancing virtual server

Receives and redirects requests to an appropriate server. Choice of the appropriate server is based on which of the various load balancing methods the user configures.

Cache redirection virtual server

Redirects client requests for dynamic content to origin servers and static content to cache servers. Cache redirection vservers often work in conjunction with load balancing vservers.

Content switching virtual server

Directs traffic to a server on the basis of the content that the client has requested. For example, you can create a content switching vserver that directs all client requests for images to a server that serves images only. Content switching vservers often work in conjunction with load balancing vservers.

Virtual private network (VPN) virtual server

Decrypts tunneled traffic and sends it to intranet applications.

SSL virtual server

Receives and decrypts SSL traffic, and then redirects to an appropriate server. Choosing the appropriate server is similar to choosing a load balancing virtual server.

Understanding Services

Services represent applications on a server. While services are normally combined with vservers, in the absence of a vserver, a service can still manage application-specific traffic. For example, you can create an HTTP service on a NetScaler to represent a web server application. When the client attempts to access a web site hosted on the web server, the NetScaler intercepts the HTTP requests and creates a transparent connection with the web server.

In service-only mode, a NetScaler functions as a proxy. It terminates client connections, uses a SNIP or MIP to establish a connection to the server, and translates incoming client requests to the SNIP or MIP. Although the clients send requests directly to the IP address of the server, the server sees them as coming from the SNIP or MIP. The NetScaler translates the IP addresses, port numbers, and sequence numbers.

A service is also a point for applying features. Consider the example of SSL acceleration. To use this feature, you must create an SSL service and bind an SSL certificate to the service. When the NetScaler receives an HTTPS request, it decrypts the traffic and sends it, in clear text, to the server. Only a limited set of features can be configured in the service-only case.

Services use entities called monitors to track the health of applications. Every service has a default monitor, which is based on the service type, bound to it. As specified by the settings configured on the monitor, the NetScaler sends probes to the application at regular intervals to determine its state. If the probes fail, the NetScaler marks the service as down. In such cases, the NetScaler responds to client requests with an appropriate error message or re-routes the request as determined by the configured load balancing policies.

Understanding Policies and Expressions

A policy defines specific details of traffic filtering and management on a NetScaler. It consists of two parts: the expression and the action. The expression defines the types of requests that the policy matches. The action tells the NetScaler what to do when a request matches the expression. As an example, the expression might be to match a specific URL pattern to a type of security attack, with the action being to drop or reset the connection. Each policy has a priority, and the priorities determine the order in which the policies are evaluated.

When a NetScaler receives traffic, the appropriate policy list determines how to process the traffic. Each policy on the list contains one or more expressions, which together define the criteria that a connection must meet to match the policy.

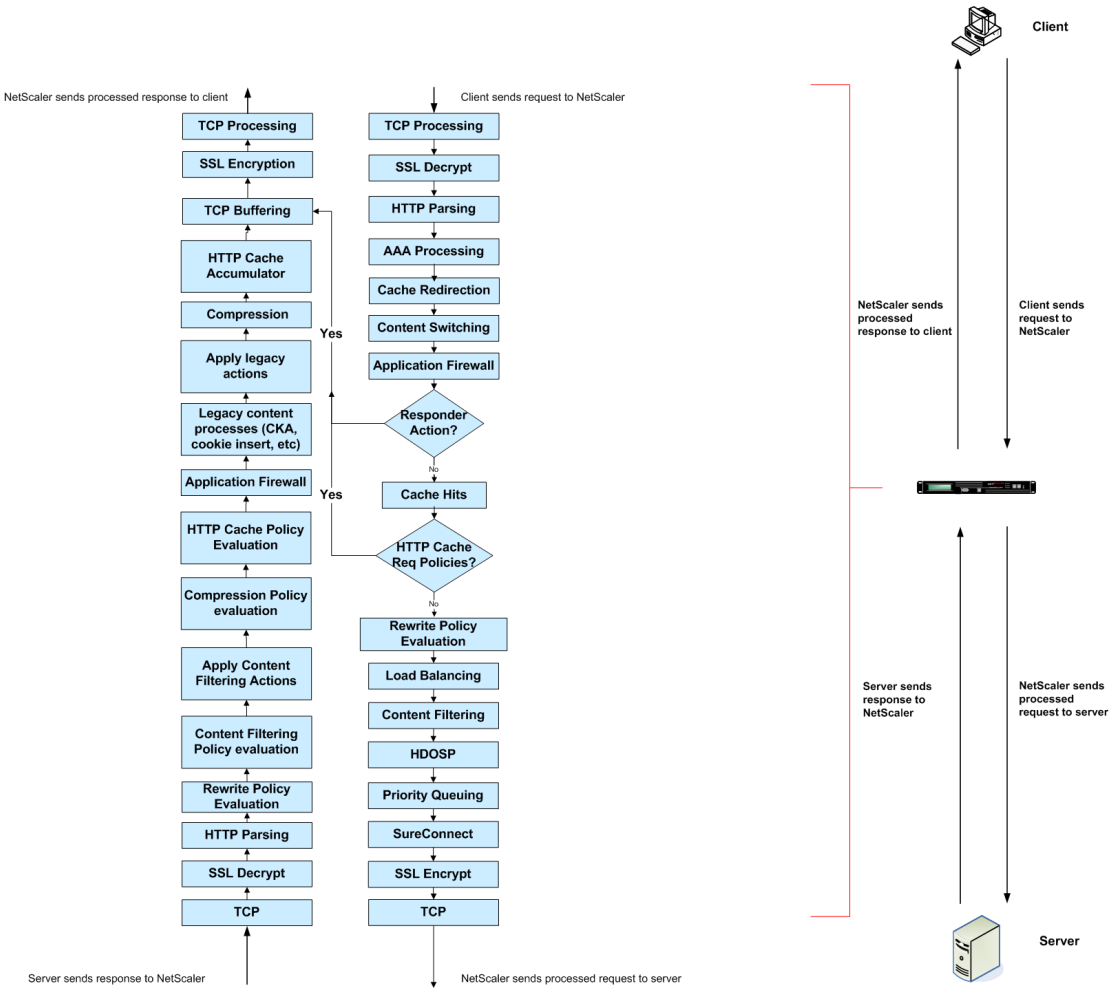
For all policy types except Rewrite policies, a NetScaler implements only the first policy that a request matches, not any additional policies that it might also match. For Rewrite policies, the NetScaler evaluates the policies in order and, in the case of multiple matches, performs the associated actions in that order. Policy priority is important for getting the results you want.

Processing Order of Features

Depending on requirements, you can choose to configure multiple features. For example, you might choose to configure both compression and SSL offload. As a result, an outgoing packet might be compressed and then encrypted before being sent to the client.

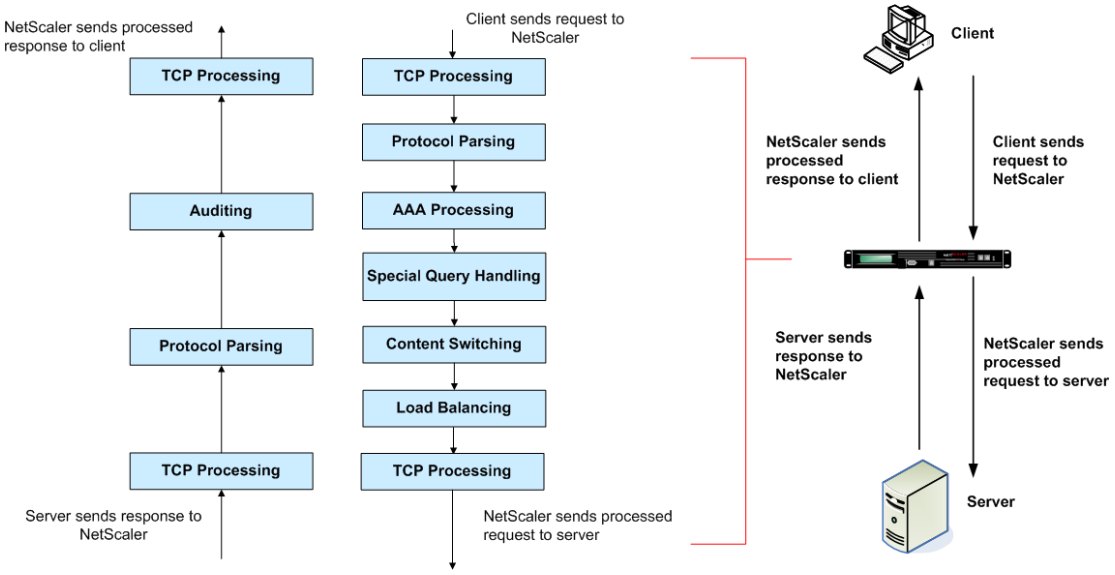
The following figure shows the L7 packet flow in the NetScaler.

Figure 1. L7 Packet Flow Diagram



The following figure shows the DataStream packet flow in the NetScaler. DataStream is supported only for MySQL databases. For information about the DataStream feature, see DataStream.

Figure 2. DataStream Packet Flow Diagram


NetScaler sends processed response to client
TCP Processing
Auditing
Protocol Parsing
TCP Processing
Server sends response to NetScaler
Client sends request to NetScaler
TCP Processing
Protocol Parsing
AAA Processing
Special Query Handling
Content Switching
Load Balancing
TCP Processing
NetScaler sends processed request to server
Client
NetScaler sends processed response to client
Client sends request to NetScaler
Server sends response to NetScaler
NetScaler sends processed request to server
Server

Introduction to the Citrix NetScaler Product Line

The Citrix® NetScaler® product line optimizes delivery of applications over the Internet and private networks, combining application-level security, optimization, and traffic management into a single, integrated appliance. You install a NetScaler appliance in your server room and route all connections to your managed servers through it. The NetScaler features that you enable and the policies you set are then applied to incoming and outgoing traffic.

A NetScaler can be integrated into any network as a complement to existing load balancers, servers, caches,and firewalls. It requires no additional client or server side software, and can be configured using the NetScaler web-based GUI and CLI configuration utilities.

NetScaler appliances are available in a variety of hardware platforms that have a range of specifications, including multicore processors.

The NetScaler operating system is the base operating system for all NetScaler hardware platforms. The NetScaler operating system is available in three editions: Standard, Enterprise, and Platinum.

Citrix NetScaler Hardware Platforms

NetScaler hardware is available in a variety of platforms that have a range of hardware specifications, including multicore processors. All hardware platforms support some combination of Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet interfaces.

The following platforms are available for NetScaler 9.3.

- Citrix NetScaler 7000
- Citrix NetScaler 9010/9010 FIPS
- Citrix NetScaler 10010
- Citrix NetScaler 12000/12000 10G
- Citrix NetScaler MPX™ 5500
- Citrix NetScaler MPX 7500/9500
- Citrix NetScaler MPX 9700/10500/12500/15500
- Citrix NetScaler MPX 15000
- Citrix NetScaler MPX 17000
- Citrix NetScaler MPX 17500/19500/21500

For more information about the hardware platform specifications, see Introduction to the Hardware Platforms.

The following tables list different editions of the NetScaler and the hardware platforms on which they are available.

Table 1. Product Editions and Standard Hardware Platforms

Hardware	7000	9010	10010	12000
Platinum Edition	Yes	Yes	Yes	Yes
Enterprise Edition	Yes	Yes	Yes	Yes
Standard Edition	Yes	Yes	No	No

Table 2. Product Editions and MPX Hardware Platforms

Hardware	MPX 5500	MPX 7500/9500	MPX 15000	MPX 17000	MPX 9700/10500/12500/15500	MPX 17500/19500/21500

Platinum Edition	Yes	Yes	Yes	Yes	Yes	Yes
Enterprise Edition	Yes	Yes	Yes	Yes	Yes	Yes
Standard Edition	Yes	Yes	Yes	Yes	Yes	Yes

Citrix NetScaler Editions

The NetScaler operating system is available in Standard, Enterprise, and Platinum editions. The Enterprise and Standard editions have limited features available. Feature licenses are required for all editions.

For instructions on how to obtain and install licenses, see NetScaler Licenses.

The Citrix NetScaler editions are described as follows:

Citrix NetScaler, Standard Edition. Provides small and medium enterprises with comprehensive Layer 4- Layer 7 (L4-L7) traffic management, enabling increased web application availability.

Citrix NetScaler, Enterprise Edition. Provides web application acceleration and advanced L4-L7 traffic management, enabling enterprises to increase web application performance and availability and reduce datacenter costs.

- *Citrix NetScaler, Platinum Edition*. Provides a web application delivery solution that reduces data center costs and accelerates application performance, with end-to-end visibility of application performance, and provides advanced application security.

The following table summarizes the features supported by each edition in the Citrix NetScaler product line:

Table 1. Citrix NetScaler Application Delivery Product Line Features

Key Features	Platinum Edition	Enterprise Edition	Standard Edition
Application availability			
Layer 4 load balancing	Yes	Yes	Yes
Layer 7 content switching	Yes	Yes	Yes
AppExpert rate controls	Yes	Yes	Yes
IPv6 support	Yes	Yes	Yes
Global server load balancing (GSLB)	Yes	Yes	Optional
Dynamic routing protocols	Yes	Yes	No
Surge protection	Yes	Yes	No
Priority queuing	Yes	Yes	No
Application acceleration			
Client and server TCP optimizations	Yes	Yes	Yes
Citrix® AppCompress™ for HTTP	Yes	Yes	Optional
Citrix® AppCache™	Yes	Optional	No
Citrix® Branch Repeater™ client	Yes	No	No

Application security			
Layer 4 DoS defenses	Yes	Yes	Yes
Layer 7 content filtering	Yes	Yes	Yes
HTTP/URL Rewrite	Yes	Yes	Yes
Citrix® Access Gateway™, EE SSL VPN	Yes	Yes	Yes
Layer 7 DoS Defenses	Yes	Yes	No
AAA security	Yes	Yes	No
Application Firewall with XML security	Yes	Optional	No
Simple manageability			
AppExpert visual policy builder	Yes	Yes	Yes
AppExpert service callouts	Yes	Yes	Yes
AppExpert templates	Yes	Yes	Yes
Role-based administration	Yes	Yes	Yes
Configuration wizards	Yes	Yes	Yes
Citrix® Command Center	Yes	Yes	No
Citrix® EdgeSight™ for NetScaler	Yes	Optional	No
Web 2.0 optimization			
Rich Internet application support	Yes	Yes	Yes
Advanced server offload	Yes	Yes	No
Lower total cost of ownership (TCO)			
TCP buffering	Yes	Yes	Yes
TCP multiplexing	Yes	Yes	Yes
SSL offload and acceleration	Yes	Yes	Yes
Cache redirection	Yes	Yes	No
Citrix® EasyCall™	Yes	No	No

Note: While we have taken care to ensure absolute accuracy when compiling this information, it might change. For the latest information, see Citrix Support at http://www.citrix.com.

Installing the NetScaler Hardware

Before installing a NetScaler® appliance, review the pre-installation checklist. A NetScaler is typically mounted in a rack, and all models ship with rack-rail hardware. All models except the 7000 support small form factor pluggable SFP, XFP, or SFP+ transceivers. After mounting the appliance and installing the transceivers, connect the NetScaler to your network. Use a console cable to connect the NetScaler to a personal computer so that you can perform an initial configuration. After connecting everything else, connect the NetScaler to a power source.

Reviewing the Pre-installation Checklist

The hardware accessories for your particular appliance, such as cables, adapters, and rail kit, will vary depending on the hardware platform you ordered. Unpack the box that contains your new appliance on a sturdy table with plenty of space and inspect the contents.

Use the following list to verify that you received everything that should have been included in the box.

- One NetScaler
- 1 Ethernet cable
- 1 RJ-45-to-RS-232 serial cable
- 1 RJ-45-to-DB-9 adapter
- The following list specifies the number of power cables included for each appliance model:
  - MPX 5500, MPX 7500/MPX 9500, and 7000: 1 power cable
  - 9010, 10010, 12000, MPX 9700/10500/12500/15500, MPX15000, MPX 17000 and MPX 17500/19500/21500: 2 power cables

  Note: Make sure that a power outlet is available for each cable.
- 1 mounting rail kit with all the models.

In addition to the items included in the box with your new appliance, you will need the following items to complete the installation and initial configuration process.

- Additional Ethernet cables for each additional Ethernet port that you will connect to your network.
- One available Ethernet port on your network switch or hub for each Ethernet port you want to connect to your network.
- A desktop or laptop/notebook computer to serve as a management workstation.

Rack Mounting the Appliance

Most appliances can be installed in standard server racks. The appliances ship with a set of rails, which you must install before you mount the appliance. The only tool you will need to install an appliance is a Phillips screwdriver.

Caution: If you are installing the appliance as the only unit in the rack, mount it at the bottom. If the rack contains other units, make sure that the heaviest unit is at the bottom. If the rack has stabilizing devices available, install them before mounting the appliance.

The 7000, MPX 5500, MPX 7500/9500 appliances each require one rack unit. The 9010, 10010, 12000, MPX 9700/10500/12500/15500, MPX 15000, MPX 17000, MPX 11500/13500/14500/16500/18500, and MPX 17500/19500/21500 appliances each require two rack units. Each of these units ships with a mounting rail kit that contains two rail assemblies, one for the left side and the other for the right side of the appliance, and screws to attach the rails. An assembly consists of an inner rail and a rack rail.

To mount the appliance, you must first install the rails and then install the appliance in the rack.

Perform the following tasks to mount the appliance:

- Remove the inner rails from the rail assembly.
- Attach the inner rails to the appliance.
- Install the rack rails on the rack.
- Install the appliance in the rack.

To remove the inner rails from the rail assembly

1. Place the rail assembly on a flat surface.
2. Slide out the inner rail toward the front of the assembly.
3. Depress the locking tabs until the inner rail comes all the way out of the rail assembly, as shown in the following figure.

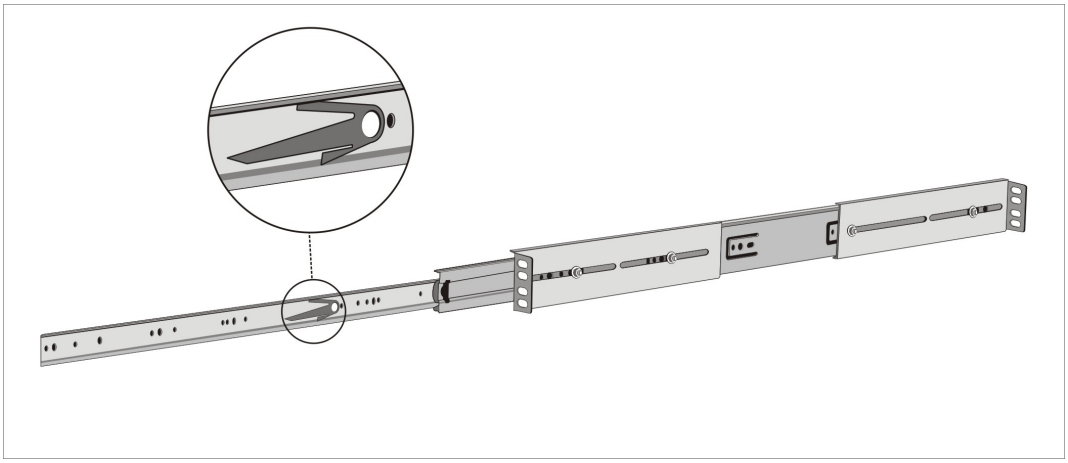


Figure 1. Removing inner rails

4. Repeat steps 1 through 3 to remove the second inner rail.

To attach the inner rails to the appliance

1. Position the right inner rail behind the ear bracket on the right side of the appliance.
2. Align the holes on the rail with the corresponding holes on the side of the appliance.
3. Attach the rail to the appliance with screws, as shown in the following figure.

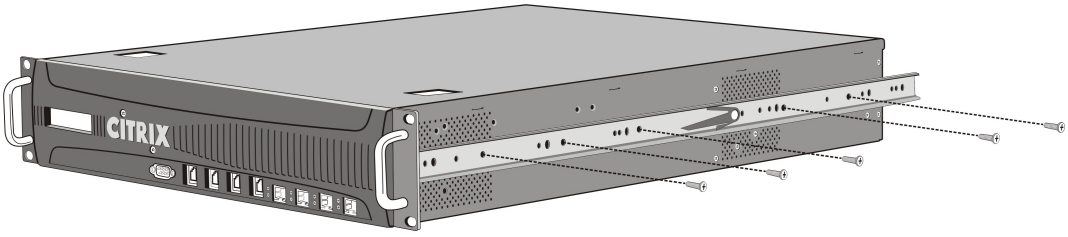


Figure 2. Attaching inner rails

4. Repeat steps 1 through 3 to install the left inner rail on the left side of the appliance.

To install the rack rails

1. Position the rack rails at the desired location in the rack, keeping the sliding rail guide facing inward.
2. Snap the tool-less rails to the rack.

   Note: Make sure that both rack rails are at same height and that the rail guides are facing inward.

To install the appliance in the rack

1. Align the inner rails, attached to the appliance, with the rack rails.
2. Slide the appliance into the rack rails, keeping the pressure even on both sides.
3. Verify that the appliance is locked in place by pulling it all the way out from the rack.

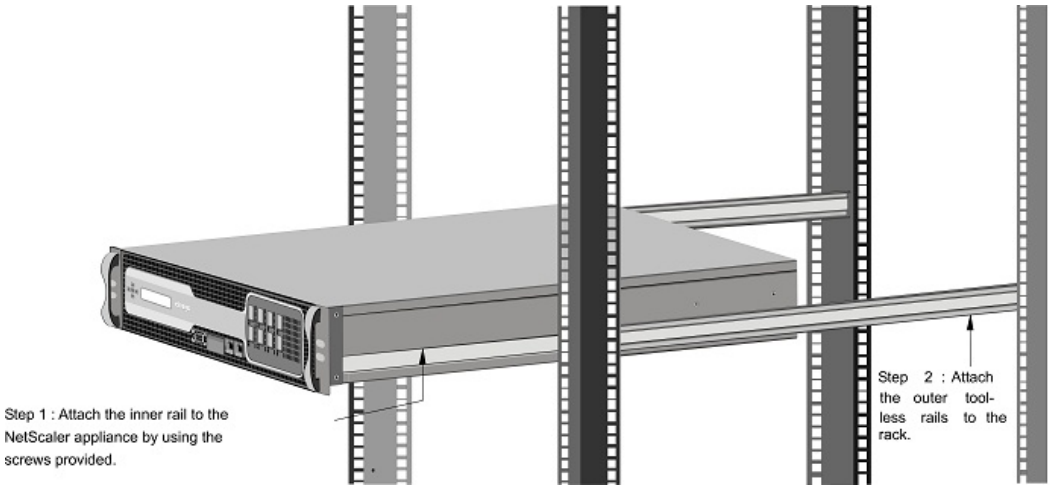


Figure 3. Rack Mounting the Appliance

Installing and Removing SFP Transceivers

Note: This section applies to the 9010, 10010, 12000, and MPX 9700/10500/12500/15500 appliances.

A Small Form Factor Pluggable (SFP) is a compact transceiver that can operate at speeds of up to 1 gigabit per second and is available in both copper and fiber types. Inserting an SFP copper transceiver converts the SFP port to a 1000BASE-T port. Inserting an SFP fiber transceiver converts the SFP port to a 1000BASE-X port. Auto-negotiation is enabled by default on the SFP port into which you insert your SFP transceiver. As soon as a link between the port and the network is established, the speed and mode are matched on both ends of the cable.

Note: SFP transceivers are hot-swappable on the 9010, 10010, 12000 appliances. However, SFP transceivers are **not hot-swappable** on the MPX 9700/10500/12500/15500 appliances. You must shutdown the MPX appliance before removing or inserting a transceiver into an SFP port on the MPX appliance.

Caution: Only SFP transceivers provided by Citrix Systems are supported on NetScaler appliances. Attempting to install third-party SFP transceivers on your NetScaler appliance voids the warranty.

Insert SFP transceivers into the SFP ports on the front panel of the appliance. Frequent installation and removal of transceivers shortens their life span. Follow the removal procedure carefully to avoid damaging the SFP transceiver or the appliance.

Caution: Do not install the transceivers with the cables attached. Doing so can damage the cable, the connector, or the optical interface of the transceiver.

To install an SFP transceiver

1. Remove the SFP transceiver carefully from its box. DANGERDo not look directly into fiberoptic transceivers or cables. They emit laser beams that can damage your eyes.

2. Align the SFP transceiver to the front of the SFP transceiver port on the front panel of the appliance, as shown in the following figure.

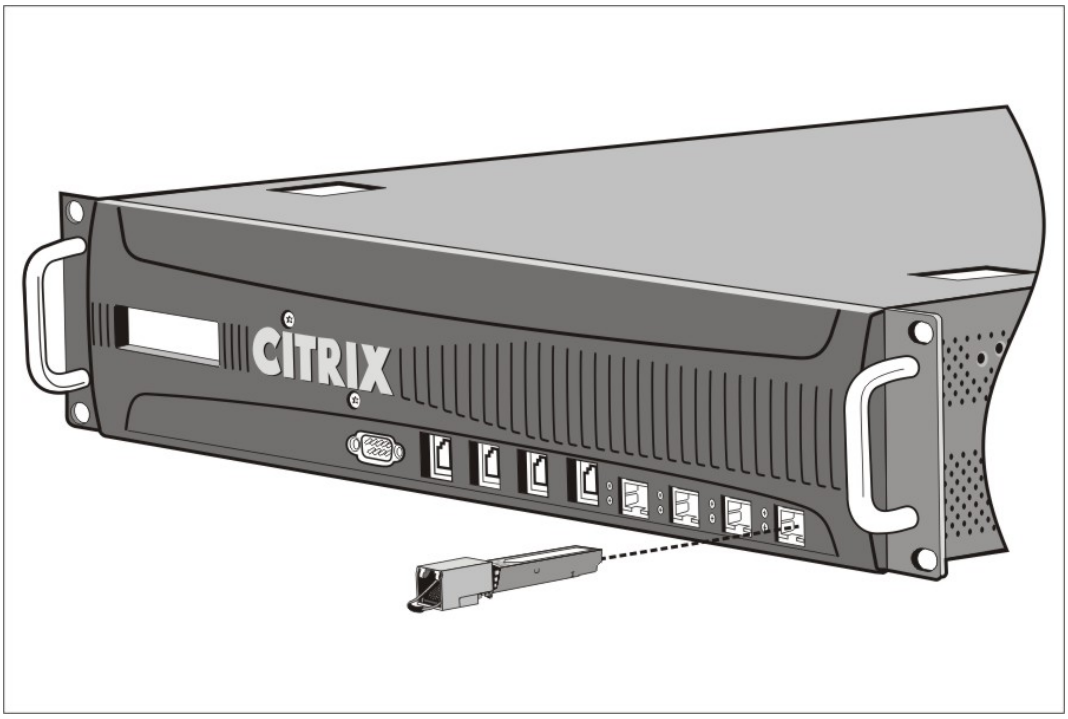


Figure 1. Installing an SFP transceiver

3. Hold the SFP transceiver between your thumb and index finger and insert it into the SFP transceiver port, pressing it in until you hear the transceiver snap into place.

4. Lock the transceiver.

5. Verify that the LED is green and blinks twice, which indicates that the transceiver is functioning correctly.

6. If you are using a fiber SFP transceiver, remove the dust caps attached to the transceiver and the cable only when you are ready to insert the cable.

To remove an SFP transceiver

1. Disconnect the cable from the SFP transceiver. If you are using a fiberoptic cable, replace the dust cap on the cable before putting it away. DANGERDo not look directly into fiberoptic transceivers or cables. They emit laser beams that can damage your eyes.

2. Unlock the SFP transceiver.

3. Hold the SFP transceiver between your thumb and index finger and slowly pull it out of the port.

4. If you are removing a fiber SFP transceiver, replace the dust cap before putting it away.

5. Put the SFP transceiver into its original box or another appropriate container.

Installing and Removing XFP and SFP+ Transceivers

Note: This section applies to the 12000 10G, MPX 9700/10500/12500/15500, MPX 15000, MPX 17000, and MPX 17500/19500/21500 appliances.

A 10-Gigabit Small Form Factor Pluggable (XFP or SFP+) is a compact optical transceiver that can operate at speeds of up to 10 gigabits per second. The 12000 10G, MPX 15000, and MPX 17000 appliances use XFP transceivers and the MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500, and MPX 17500/19500/21500 appliances use SFP+ transceivers. Auto-negotiation is enabled by default on the XFP/SFP+ ports into which you insert your XFP/SFP+ transceiver. As soon as a link between the port and the network is established, the mode is matched on both ends of the cable and for SFP+ transceivers, the speed is also autonegotiated.

Note: XFP and SFP+ transceivers are **not hot-swappable** on the NetScaler appliances. You must restart a NetScaler appliance after you insert a 10 GE XFP or SFP+ transceiver.

Caution: Only XFP/SFP+ transceivers provided by Citrix Systems are supported on NetScaler appliances. Attempting to install third-party XFP/SFP+ transceivers on your NetScaler appliance voids the warranty.

Insert the XFP/SFP+ transceivers into the XFP/SFP+ ports on the front panel of the appliance. Frequent installation and removal of transceivers shortens their life span. Follow the removal procedure carefully to avoid damaging the transceiver or the appliance.

Caution: Do not install the transceivers with the cables attached. Doing so can damage the cable, the connector, or the optical interface of the transceiver.

To install an XFP/SFP+ transceiver

1. Remove the XFP/SFP+ transceiver carefully from its box. DANGERDo not look directly into fiberoptic transceivers and cables. They emit laser beams that can damage your eyes.

2. Align the XFP/SFP+ transceiver to the front of the XFP/SFP+ transceiver port on the front panel of the appliance.

3. Hold the XFP/SFP+ transceiver between your thumb and index finger and insert it into the XFP/SFP+ transceiver port, pressing it in until you hear the transceiver snap into place.

4. Move the locking hinge to the DOWN position as shown in the following figure.

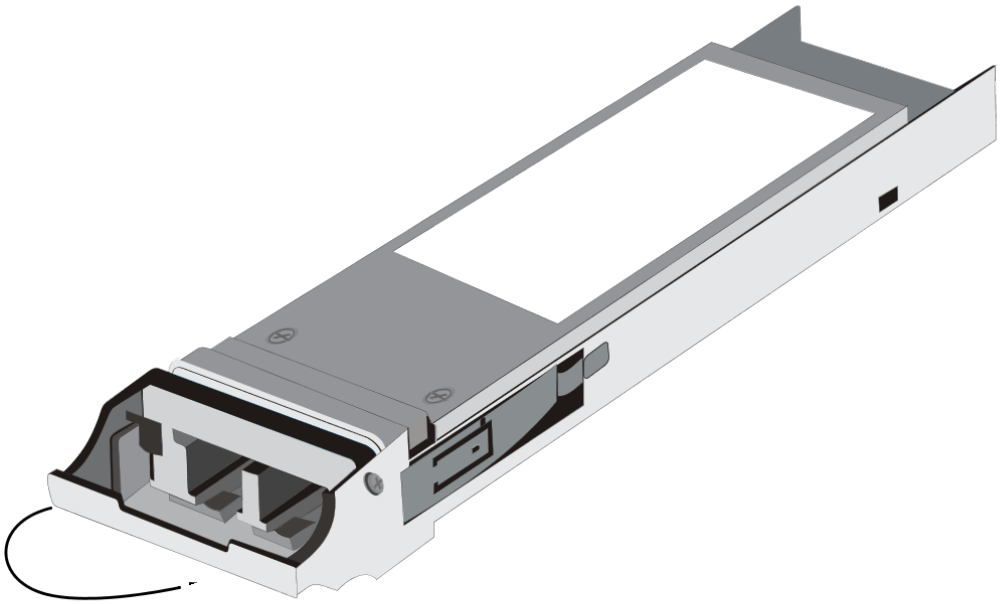


Figure 1. Locking an XFP transceiver

5. Verify that the LED is green and blinks twice, which indicates that the transceiver is functioning correctly.

6. Remove the dust caps attached to the transceiver and cable only when you are ready to insert the cable.

To remove an XFP/SFP+ transceiver

1. Disconnect the cable from the XFP/SFP+ transceiver. Replace the dust cap on the cable before putting it away. DANGERDo not look directly into fiberoptic transceivers or cables. They emit laser beams that can damage your eyes.

2. Unlock the XFP/SFP+ transceiver by moving the locking hinge to the UP position.

3. Hold the XFP/SFP+ transceiver between your thumb and index finger and slowly pull it out of the port.

4. Replace the dust cap on the transceiver before putting it away.

5. Put the XFP/SFP+ transceiver into its original box or another appropriate container.

Connecting the Cables

When the appliance is securely mounted on the rack, you are ready to connect the cables. Ethernet cables and the optional console cable are connected first. Connect the power cable last.

DANGERRemove all jewelry and other metal objects that might come in contact with power sources or wires before installing or repairing the appliance. When you touch both a live power source or wire and ground, any metal objects can heat up rapidly, and may cause burns, set clothing on fire, or fuse the metal object to an exposed terminal.

Connecting the Ethernet Cables

Ethernet cables connect your appliance to the network. The type of cable you need depends on the type of port used to connect to the network. Use a category 5e or category 6 Ethernet cable with a standard RJ-45 connector on a 10/100/1000BASE-T port or 1-gigabit SFP copper transceiver. Use a fiber optic cable with an LC duplex connector with an SFP fiber transceiver, SFP+, or XFP transceiver. The type of connector at the other end of the fiber optic cable depends on the port of the device that you are connecting to.

To connect an Ethernet cable to a 10/100/1000BASE-T port or 1-gigabit SFP copper transceiver

1. Insert the RJ-45 connector on one end of your Ethernet cable into an appropriate port on the front panel of the appliance as shown in the following figure.

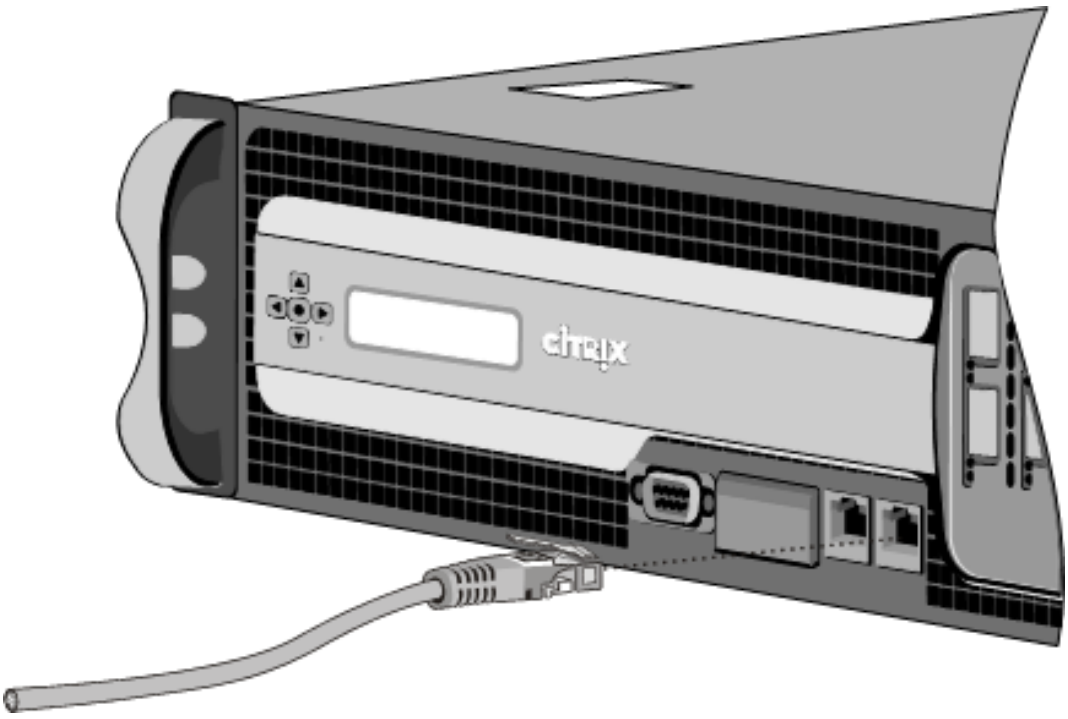


Figure 1. Inserting an Ethernet cable

2. Insert the RJ-45 connector on the other end into the target device, such as a router or switch.

3. Verify that the LED glows amber when the connection is established.

To connect the Ethernet cable to an SFP fiber, SFP+, or XFP transceiver

1. Remove the dust caps from the transceiver and cable.

2. Insert the LC connector on one end of the fiber optic cable into the appropriate port on the front panel of the appliance.

3. Insert the connector on the other end into the target device, such as a router or switch.

4. Verify that the LED glows amber when the connection is established.

Connecting the Console Cable

You can use the console cable to connect your appliance to a computer or terminal from which you can configure the appliance. Alternatively, you can use a computer connected to the network. Before connecting the console cable, configure the computer or terminal to support VT100 terminal emulation, 9600 baud, 8 data bits, 1 stop bit, parity, and flow control set to NONE. Then connect one end of the console cable to the RS232 serial port on the appliance and the other end to the computer or terminal.

To connect the console cable to a computer or terminal

1. Insert the DB-9 connector at the end of the cable into the console port that is located on the front panel of the appliance as shown in the following figure.

   Figure 2. Inserting a console cable

   Note: To use a cable with an RJ-45 converter, insert the optional converter provided into the console port and attach the cable to it.

2. Insert the RJ-45 connector at the other end of the cable into the serial port of the computer or terminal.

Connecting the Power Cable

An MPX 5500, MPX 7500/9500, and 7000 appliance has one power cable. All the other appliances come with two power cables, but they can also operate if only one power cable is connected. A separate ground cable is not required, because the three-prong plug provides grounding.

To connect the appliance to the power source

1. Connect one end of the power cable to the power outlet on the back panel of the appliance, next to the power supply, as shown in the following figure.

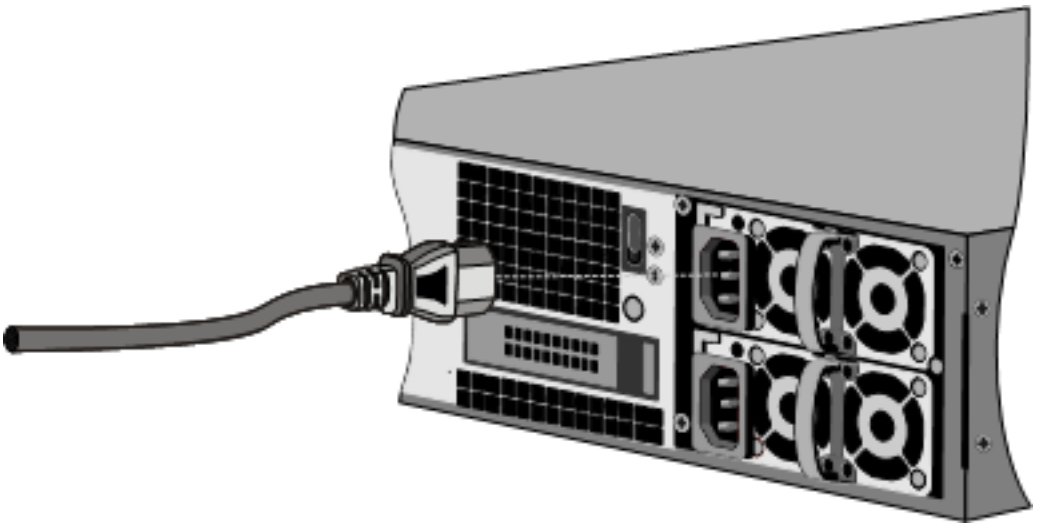


Figure 3. Inserting a power cable

2. Connect the other end of the power cable to a standard 110V/220V power outlet.

3. If a second power supply is provided, repeat steps 1 and 2 to connect the second power supply.

Note: The 9010, 10010, 12000, MPX 9700/10500/12500/15500, and MPX 17500/19500/21500 appliances emit a high-pitched alert if one power supply fails or if you connect only one power cable to the appliance. To silence the alarm, you can press the small red button located on the back panel of the appliance.

Accessing a Citrix NetScaler

A NetScaler® appliance has both a command line interface (CLI) and a graphical user interface (GUI). The GUI includes a configuration utility for configuring the appliance and a statistical utility, called Dashboard. For initial access, all NetScaler appliances ship with the default NetScaler IP address (NSIP) of 192.168.100.1 and default subnet mask of 255.255.0.0. You can assign a new NSIP and an associated subnet mask during initial configuration.

If you encounter an IP address conflict when deploying multiple NetScaler units, check for the following possible causes:

- Did you select an NSIP that is an IP address already assigned to another device on your network?
- Did you assign the same NSIP to multiple NetScaler appliances?
- The NSIP is reachable on all physical ports. The ports on a NetScaler are host ports, not switch ports.

The following table summarizes the available access methods.

Table 1. Methods for Accessing a NetScaler

Access Method	Port	Default IP Address Required? (Y/N)
CLI	Console	N
CLI and GUI	Ethernet	Y

Using the Command Line Interface

You can access the CLI either locally, by connecting a workstation to the console port, or remotely, by connecting through secure shell (SSH) from any workstation on the same network.

For more information about the features of the CLI, including SSH, see the *Citrix NetScaler Command Reference Guide* at http://support.citrix.com/article/CTX128678.

Logging on to the Command Line Interface through the Console Port

The NetScaler has a console port for connecting to a computer workstation. To log on to the NetScaler, you need a serial crossover cable and a workstation with a terminal emulation program.

To log on to the CLI through the console port

1. Connect the console port to a serial port on the workstation, as described in Connecting the Console Cable .
2. On the workstation, start HyperTerminal or any other terminal emulation program. If the logon prompt does not appear, you may need to press ENTER one or more times to display it.
3. Log on as `nsroot`. For initial configuration, use `nsroot` as the administrative password. For subsequent access, use the password assigned during initial configuration. The command prompt (>) appears on the workstation monitor.

Logging on to the Command Line Interface by using SSH

The SSH protocol is the preferred remote access method for accessing a NetScaler remotely from any workstation on the same network. You can use either SSH version 1 (SSH1) or SSH version 2 (SSH2.)

If you do not have a working SSH client, you can download and install any of the following SSH client programs:

- PuTTY

Open Source software supported on multiple platforms. Available at:

http://www.chiark.greenend.org.uk/~sgtatham/putty/

- Vandyke Software SecureCRT

  Commercial software supported on the Windows platform. Available at:

  http://www.vandyke.com/products/securecrt/

These programs have been tested by the Citrix NetScaler team, which has verified that they work correctly with a NetScaler. Other programs may also work correctly, but have not been tested.

To verify that the SSH client is installed properly, use it to connect to any device on your network that accepts SSH connections.

To log on to a NetScaler by using an SSH client

1. On your workstation, start the SSH client.
2. For initial configuration, use the default NetScaler IP address (NSIP), which is 192.168.100.1. For subsequent access, use the NSIP that was assigned during initial configuration. Select either SSH1 or SSH2 as the protocol.
3. Log on as `nsroot`. For initial configuration, use `nsroot` as the administrative password. For subsequent access, use the password assigned during initial configuration. For example:

```
login as: nsroot
Using keyboard-interactive authentication.
Password:
Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9

 Done
>
```

Using the Graphical User Interface

The graphical user interface includes a configuration utility and a statistical utility, called Dashboard, either of which you access through a workstation connected to an Ethernet port on the NetScaler. If your computer does not have a supported Java plug-in installed, the utillity prompts you to download and install the plug-in the first time you log on. If automatic installation fails, you can install the plug-in separately before you attempt to log on to the configuration utility or Dashboard.

The system requirements for the workstation running the GUI are as follows:

- For Windows-based workstations, a Pentium® 166 MHz or faster processor with at least 48 MB of RAM is recommended for applets running in a browser using a Java plug-in product. You should have 40 MB free disk space before installing the plug-in.
- For Linux-based workstations, a Pentium platform running Linux kernel v2.2.12 or above, and glibc version 2.12-11 or later. A minimum of 32 MB RAM is required, and 48 MB RAM is recommended. The workstation should support 16-bit color mode, KDE and KWM window managers used in conjunction, with displays set to local hosts.
- For Solaris-based workstations, a Sun running either Solaris 2.6, Solaris 7, or Solaris 8, and the Java 2 Runtime Environment, Standard Edition, version 1.6 or later.

Your workstation must have a supported web browser and version 1.6 or above of the Java® applet plug-in installed to access the configuration utility and Dashboard.

The following browsers are supported.

On the Windows 7 operating system:

- Internet Explorer 8
- Google Chrome 4.1

On the Windows XP Home and Windows XP Professional operating systems:

- Internet Explorer 6, 7, 8
- Netscape 7.0
- Google Chrome 4.1

On the Red Hat Linux 5.0 and 6.0 operating systems:

- Mozilla Firefox 1.0, 2.0, 3.0

Using the Configuration Utility

Once you log on to the configuration utility, you can configure the NetScaler through a graphical interface that includes context-sensitive help.

If your computer does not have a supported Java plug-in installed, the first time you log on to the NetScaler, the configuration utility will prompt you to download and install the plug-in.

Note: Prior to installing the Java 2 Runtime Environment, ensure that you have installed the full set of required operating system patches needed for the current Java release.

To log on to the configuration utility

1. Open your web browser and enter the NetScaler IP (NSIP) as an HTTP address. If you have not yet set up the initial configuration, enter the default NSIP (`http://192.168.100.1`). The **Citrix Logon** page appears.

   Note: If you have two NetScaler appliances in a high availability setup, make sure that you do not access the GUI by entering the IP address of the secondary NetScaler. If you do so and use the GUI to configure the secondary NetScaler, your configuration changes will not be applied to the primary NetScaler.

2. In the **User Name** text box, type `nsroot`.

3. In the **Password** text box, type the administrative password you assigned to the nsroot account during initial configuration.

4. In the **Start in** list, click **Configuration**, and then click **Login**. The **Configuration Utility** page appears.

   Note: If your workstation does not already have a supported version of the Java runtime plug-in installed, the NetScaler prompts you to download the Java Plug-in. After the download is complete, the configuration utility page appears.

   If you need to access the online help, select Help from the Help menu at the top right corner.

Using the Statistical Utility

Dashboard, the statistical utility, is a browser-based application that displays charts and tables on which you can monitor the performance of a NetScaler.

To log on to Dashboard

1. Open your web browser and enter the NSIP as an HTTP address (http://<NSIP>). The **Citrix Logon** page appears.

2. In the **User Name** text box, type `nsroot`.

3. In the **Password** text box, type the administrative password you assigned to the nsroot account during initial configuration.

4. In the **Start in** list, click **Dashboard**, and then click **Login**.

   Note: If your workstation does not already have a supported version of the Java runtime plug-in installed, the NetScaler prompts you to download the Java Plug-in. After the download is complete, Dashboard is displayed. If automatic installation of the Java plug-in fails, you can install the plug-in separately before you attempt to log on to Dashboard.

Installing the Java Runtime Plug-in

If automatic installation of the Java plug-in fails, you can install the plug-in separately before you attempt to log on to the configuration utility or Dashboard.

Note: Before installing the Java 2 Runtime Environment, make sure that you have installed the full set of required operating system patches needed for the current Java release.

To install the Java runtime plug-in on your workstation

1. In your web browser, enter the NSIP and port number of your NetScaler: `http://<NSIP>:80` The Java plug-in icon appears.

2. Click the Java plug-in icon and follow the screen prompts to copy the plug-in installer to your workstation hard disk. The Java plug-in setup icon (for example, **j2re-1.6.0**) appears on your computer at the location you specified.

3. Double-click the plug-in setup icon, and follow the screen prompts to install the plug-in.

4. Return to your web browser and click the Java plug-in icon a second time to display the GUI logon screen.

Configuring a NetScaler for the First Time

Your new NetScaler is preconfigured with a default IP address (the NSIP) and associated subnet mask for management access. The default NSIP is 192.168.100.1 and the subnet mask (netmask) is 255.255.0.0. You can change these values to fit the addressing scheme for your network. For your initial configuration, you must also specify at least one SNIP or MIP. Before saving your new configuration, you should change the administrator password.

If you are setting up two NetScaler appliances as a high availability pair, you configure one as primary and the other as secondary.

Using the LCD Keypad

Note: The following section is applicable to MPX 5500, MPX 7500/9500, MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500, and MPX 17500/19500/21500 appliances only.

When you first install the appliance, you can configure the initial settings by using the LCD keypad on the front panel of the appliance. The keypad interacts with the LCD display module, which is also on the front panel of these appliances.

Note: You can use the LCD keypad for initial configuration on a new appliance with the default configuration. The configuration file (ns.conf) should contain the following command and default values.

set ns config -**IPAddress** 192.168.100.1 **-netmask** 255.255.0.0

The functions of the different keys are explained in the following table.

Table 1. LCD Key Functions

Key	Function
<	Moves the cursor one digit to the left.
>	Moves the cursor one digit to the right.
^	Increments the digit under the cursor.
v	Decrements the digit under the cursor.
.	Processes the information, or terminates the configuration, if none of the values are changed. This key is also known as the ENTER key.

You are prompted to enter the subnet mask, NetScaler IP address (NSIP), and gateway in that order respectively. The subnet mask is associated with both the NSIP and default gateway IP address. The NSIP is the IPv4 address of the NetScaler appliance. The default gateway is the IPv4 address for the router, which will handle external IP traffic that the NetScaler cannot otherwise route. The NSIP and the default gateway should be on the same subnet.

If you enter a valid value for the subnet mask, such as 255.255.255.224, you are prompted to enter the IP address. Similarly, if you enter a valid value for the IP address, you are prompted to enter the gateway address. If the value you entered is invalid, the following error message appears for three seconds, where xxx.xxx.xxx.xxx is the IP address you entered, followed by a request to re-enter the value.

Invalid addr!
xxx.xxx.xxx.xxx

If you press the ENTER (.) key without changing any of the digits, the software interprets this as a user exit request.The following message will be displayed for three seconds.

Exiting menu...
xxx.xxx.xxx.xxx

If all the values entered are valid, when you press the ENTER key, the following message appears.

Values accepted,
Rebooting...

The subnet mask, NSIP, and gateway values are saved in the configuration file.

Configuring a NetScaler by Using the Command Line Interface

When you first install the appliance, you can configure the inital settings by using the serial console. Connect a serial cable to the port on the appliance and the other end to a computer, as described in Connecting the Console Cable . For remote access to the command-line interface (CLI), see Logging on to the Command Line Interface by using SSH. At the CLI, you can setup or change the NSIP, subnet or mapped IP address, advanced network settings, and time zone.

To configure a NetScaler by using the command line interface

1. Connect a workstation to the NetScaler.
2. Run the vt100 terminal emulation program of your choice on your workstation or notebook computer to connect to the appliance.
   - For Microsoft Windows, you can use Hyperterminal, which is installed with all current versions of Windows.
   - For Apple Macintosh OSX, you can use the GUI-based Terminal program or the shell-based telnet client.

     Note: OSX is based on the FreeBSD UNIX platform. Most standard UNIX shell programs are available from the OSX command line.
   - For UNIX-based workstations, you can use the shell-based telnet client or any supported terminal emulation program.
3. Press ENTER. The terminal screen displays the Logon prompt.

   Note: You might have to press ENTER two or three times, depending on which terminal program you are using.
4. Log on to the appliance by using the administrator credentials.

   Note: Your sales representative or Citrix Customer Service can provide the administrator credentials.
5. At the NetScaler command prompt, you can type `config ns` and follow the prompts to complete the initial configuration. Alternatively, type the commands shown in the following steps.

   Note: To prevent an attacker from breaching your ability to send packets to the appliance, choose a non-routable IP address on your organization's LAN as your appliance IP address.
6. `set ns config` **`-ipaddress`** `<IPAddress>` **`-netmask`** `<Netmask>`
7. `add ns ip <IPAddress> <Netmask>` **`-type`** `<Type>`

8. `add route <Network> <Netmask> <Gateway>`
9. `set system user nsroot <Password>`
10. `save ns config`
11. `reboot`

Example

set ns config - ipaddress 10.102.29.60 - netmask 255.255.255.0
add ns ip 10.102.29.61 255.255.255.0 -type snip
add route 0.0.0.0 0.0.0.0 10.102.29.1
set system user nsroot administrator
save ns config
reboot

Configuring a NetScaler by Using the Configuration Utility

The configuration utility is accessed from a web browser.To configure the NetScaler using the Setup Wizard in the configuration utility, you need an administrative workstation or laptop configured on the same network as the appliance. You also need Java RunTime Environment (JRE) version 1.6 or later. You can use the Setup Wizard to configure the following initial settings:

- System IP address and subnet mask
- Subnet or Mapped IP address and subnet mask
- Host name
- Default gateway
- Time zone
- Licenses
- Administrator password

Important: Before running the Setup Wizard, you should download your licenses from the Citrix web site and put them in a location on your workstation or laptop hard drive or another device where you can access them from your web browser during configuration.

To configure initial settings by using the Setup Wizard

1. In a web browser, type `http:// 192.168.100.1`.

   Note: The operating system is preconfigured with a default IP address and associated netmask. The default IP address is 192.168.100.1 and the default netmask is 255.255.0.0.

2. In **User Name** and **Password**, type the administrator credentials. You can obtain the initial user name and password from your sales representative or from Citrix Customer Service.

3. In **Start in**, select **Configuration**, and then click **Login**.

4. In the **Setup Wizard**, click **Next**, and then follow the instructions in the wizard.

   Note: To prevent an attacker from breaching your ability to send packets to the appliance, choose a non-routable IP address on your organization's LAN as your appliance IP address.

Configuring a NetScaler by Using the XML API

You can use an external Application Programming Interface (API) to configure the NetScaler. The API allows you to create custom client applications to configure and monitor the state of the NetScaler. It is based on Simple Object Access Protocol (SOAP) over HTTP. You can download the API documentation from the Downloads page of the configuration utility.

Configuring a High Availability Pair for the First Time

You can deploy two NetScaler appliances in a high availability configuration, where one unit actively accepts connections and manages servers while the secondary unit monitors the first. The NetScaler that is actively accepting connections and managing the servers is called a primary unit and the other one is called a secondary unit in a high availability configuration. If there is a failure in the primary unit, the secondary unit becomes the primary and begins actively accepting connections.

Each NetScaler in a high availability pair monitors the other by sending periodic messages, called heartbeat messages or health checks, to determine the health or state of the peer node. If a health check for a primary unit fails, the secondary unit retries the connection for a specific time period. (For more information about high availability, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.) If a retry does not succeed by the end of the specified time period, the secondary unit takes over for the primary unit in a process called failover. The following figure shows two high availability configurations, one in one-arm mode and the other in two-arm mode.

Figure 1. High availability in one-arm mode

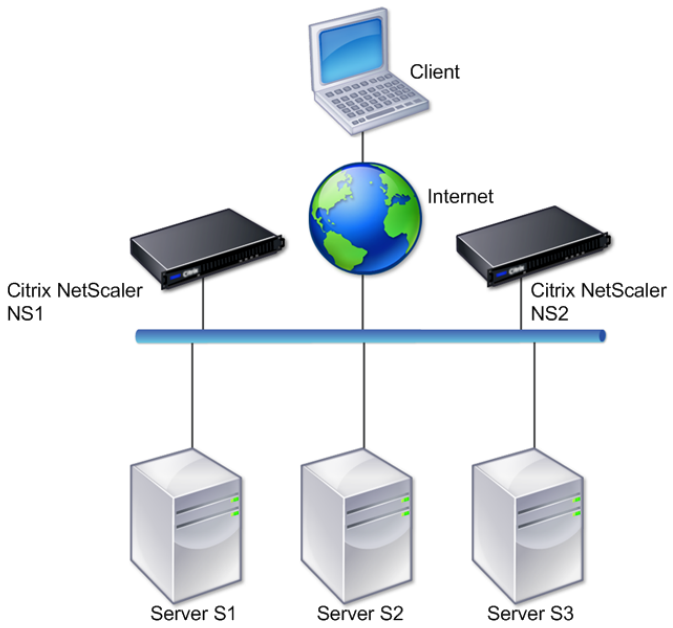
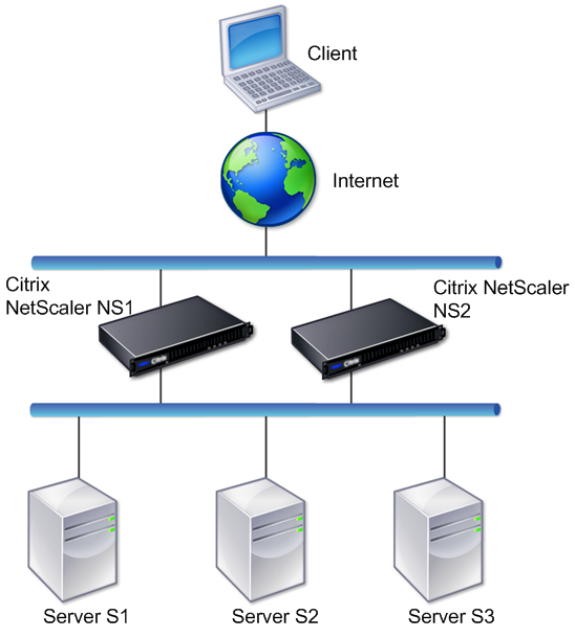


Figure 2. High availability in two-arm mode



In one-arm configuration, both NS1 and NS2 and servers S1, S2, and S3 are connected to the switch.

In two-arm configuration, both NS1 and NS2 are connected to two switches. The servers S1, S2, and S3 are connected to the second switch. The traffic between client and the servers passes through either NS1 or NS2.

To set up a high availability environment, configure one NetScaler as primary and another as secondary. Perform the following tasks on each of the NetScalers:

- Add a node.
- Disable high availability monitoring for unused interfaces.

Adding a Node

A node is a logical representation of a peer NetScaler. It identifies the peer unit by ID and NSIP. A NetScaler uses these parameters to communicate with the peer and track its state. When you add a node, the primary and secondary units exchange heartbeat messages asynchronously. The node ID is an integer that must not be greater than 64.

To add a node by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a node and verify that the node has been added:

- `add HA node <id> <IPAddress>`
- `show HA node <id>`

  Example

  ```
  add HA node 0 10.102.29.170
   Done
  > show HA node 0
  1)      Node ID:      0
          IP:   10.102.29.200 (NS200)
          Node State: UP
          Master State: Primary
          SSL Card Status: UP
          Hello Interval: 200 msecs
          Dead Interval: 3 secs
          Node in this Master State for: 1:0:41:50 (days:hrs:min:sec)
  ```

To add a node by using the configuration utility

1. In the navigation pane, expand **System** and click **High Availability**. The **High Availability** page appears.
2. On the **High Availability** page, select the **Nodes** tab.
3. Click **Add**. The **High Availability Setup** dialog box appears.
4. In the **High Availability Setup** dialog box, in the **Remote Node IP Address** text box, type an IP Address (for example, `10.102.29.170`).
5. Ensure that the **Configure remote system to participate in High Availability setup** check box is selected. By default, this check box is selected.
6. Select the **Turn off HA monitor on interfaces/channels that are down** check box to disable the HA monitor on interfaces that are down. By default, this check box is selected.
7. Verify that the node you added appears in the list of nodes under the **Nodes** tab.

Disabling High Availability Monitoring for Unused Interfaces

The high availability monitor is a virtual entity that monitors an interface. You must disable the monitor for interfaces that are not connected or being used for traffic. When the monitor is enabled on an interface whose status is DOWN, the state of the node becomes NOT UP. In a high availability configuration, a primary node entering a NOT UP state might cause a high availability failover. An interface is marked DOWN under the following conditions:

- The interface is not connected
- The interface is not working properly
- The cable connecting the interface is not working properly

To disable the high availability monitor for an unused interface by using the command line

At the NetScaler command prompt, type the following commands to disable the high availability monitor for an unused interface and verify that it is disabled:

- `set interface <id> -haMonitor OFF`
- `show interface <id>`

Example

```
> set interface 1/8 -haMonitor OFF
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4000 <ENABLED, DOWN, down, autoneg, 802.1q>
        MTU=1514, native vlan=1, MAC=00:d0:68:15:fd:3d, downtime 238h55m44s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
                throughput 0

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
```

When the high availability monitor is disabled for an unused interface, the output of the show interface command for that interface does not include "HAMON."

To disable the high availability monitor for unused interfaces by using the configuration utility

1. In the navigation pane, expand **Network** and click **Interfaces**. The **Interfaces** page appears.
2. Select the interface for which the monitor must be disabled.
3. Click **Open**. The **Modify Interface** dialog box appears.
4. In **HA Monitoring**, select the **OFF** option.
5. Click **OK**.
6. Verify that, when the interface is selected, "HA Monitoring: OFF" appears in the details at the bottom of the page.

Understanding Common Network Topologies

As described in Physical Deployment Modes, you can deploy the Citrix® NetScaler® appliance either inline between the clients and servers or in one-arm mode. Inline mode uses a two-arm topology, which is the most common type of deployment.

Setting Up Common Two-Arm Topologies

In a two-arm topology, one network interface is connected to the client network and another network interface is connected to the server network, ensuring that all traffic flows through the NetScaler. This topology might require you to reconnect your hardware and also might result in a momentary downtime. The basic variations of two-arm topology are multiple subnets, typically with the NetScaler on a public subnet and the servers on a private subnet, and transparent mode, with both the NetScaler and the servers on the public network.

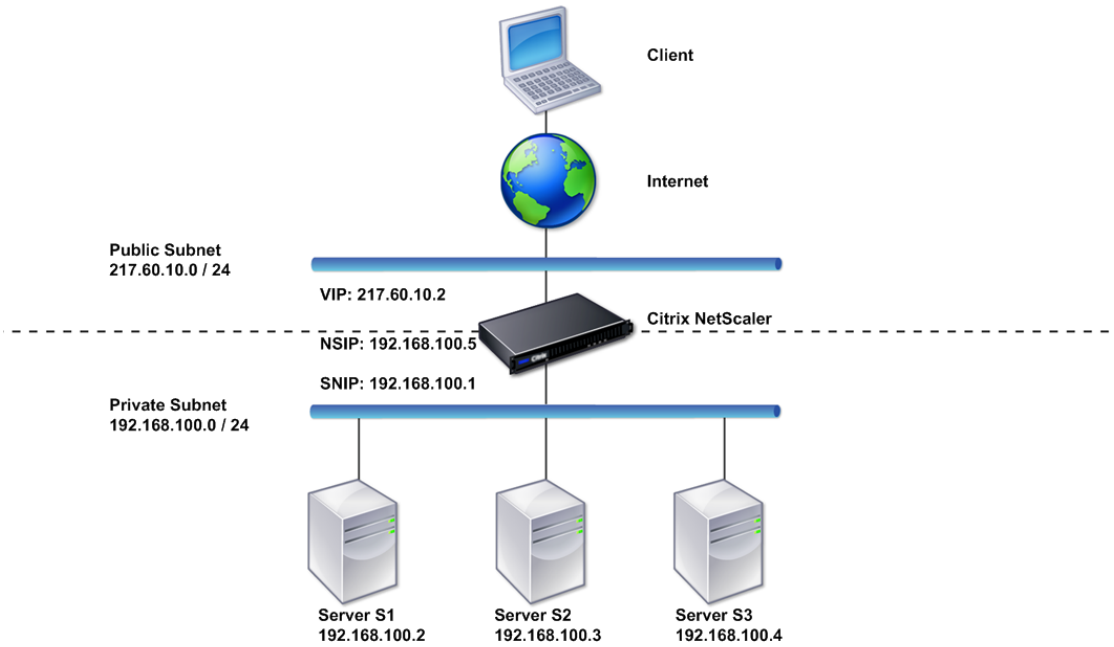
Setting Up a Simple Two-Arm Multiple Subnet Topology

One of the most commonly used topologies has the NetScaler inline between the clients and the servers, with a vserver configured to handle the client requests. This configuration is used when the clients and servers reside on different subnets. In most cases, the clients and servers reside on public and private subnets, respectively.

For example, consider a NetScaler deployed in two-arm mode for managing servers S1, S2, and S3, with a vserver of type HTTP configured on the NetScaler, and with HTTP services running on the servers. The servers are on a private subnet and a SNIP is configured on the NetScaler to communicate with the servers. The Use SNIP (USNIP) option must be enabled on the NetScaler so that it uses the SNIP instead of the MIP.

As shown in the following figure, the VIP and a SNIP are on public subnet 217.60.10.0, and the NSIP, the servers, and another SNIP are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for Two-Arm Mode, Multiple Subnets



Task overview: To deploy a NetScaler in two-arm mode with multiple subnets

1. Configure the NSIP and default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.

2. Configure the SNIP, as described in the Configuring Subnet IP Addresses (SNIPs).

3. Enable the USNIP option, as described in the To enable or disable USNIP mode by using the NetScaler command line.

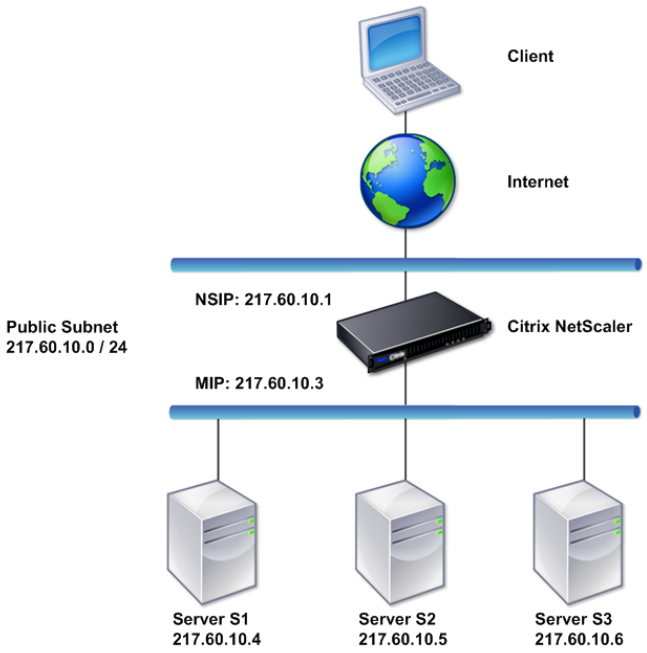
4. Configure the vserver and the services, as described in the Creating a Virtual Server and Configuring Services.

5. Connect one of the network interfaces to a private subnet and the other interface to a public subnet.

Setting Up a Simple Two-Arm Transparent Topology

Use transparent mode if the clients need to access the servers directly, with no intervening vserver. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a NetScaler is placed between the client and the server, so the traffic must pass through the NetScaler. You must enable L2 mode for bridging the packets. The NSIP and MIP are on the same public subnet, 217.60.10.0/24.

Figure 1. Topology Diagram for Two-Arm, Transparent Mode



Task overview: To deploy a NetScaler in two-arm, transparent mode

1. Configure the NSIP, MIP, and default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.
2. Enable L2 mode, as described in the Enabling and Disabling Layer 2 Mode.
3. Configure the default gateway of the managed servers as the MIP.
4. Connect the network interfaces to the appropriate ports on the switch.

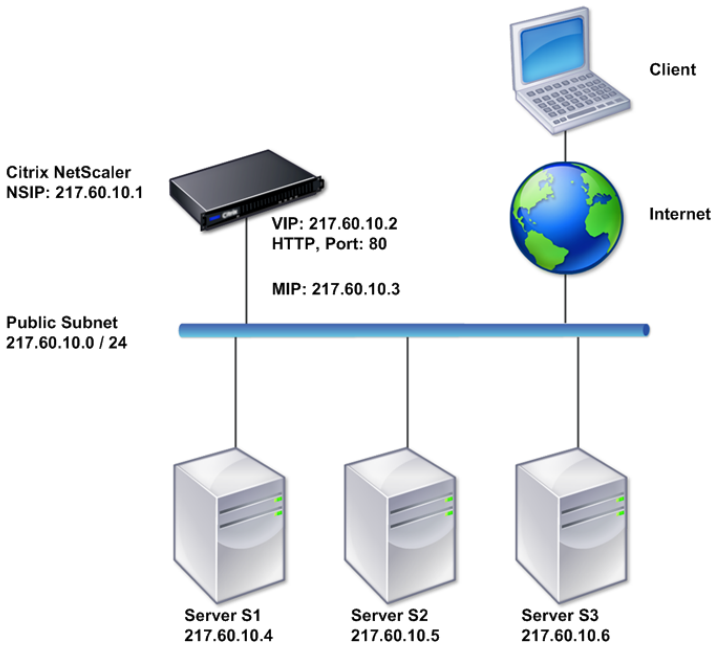
Setting Up Common One-Arm Topologies

The two basic variations of one-arm topology are with a single subnet and with multiple subnets.

Setting Up a Simple One-Arm Single Subnet Topology

You can use a one-arm topology with a single subnet when the clients and servers reside on the same subnet. For example, consider a NetScaler deployed in one-arm mode for managing servers S1, S2, and S3. A vserver of type HTTP is configured on a NetScaler, and HTTP services are running on the servers. As shown in the following figure, the NetScaler IP address (NSIP), the Mapped IP address (MIP), and the server IP addresses are on the same public subnet, 217.60.10.0/24.

Figure 1. Topology Diagram for One-Arm Mode, Single Subnet



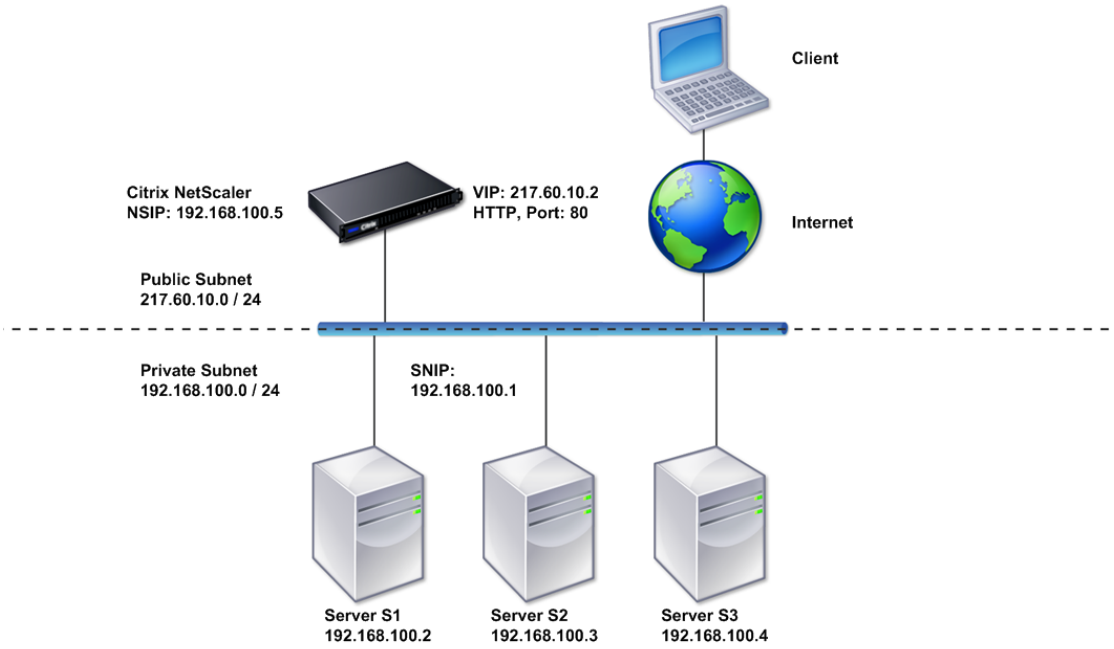
Task overview: To deploy a NetScaler in one-arm modewith a single subnet

1. Configure the NSIP, MIP, and the default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.
2. Configure the vserver and the services, as described in Creating a Virtual Server and Configuring Services.
3. Connect one of the network interfaces to the switch.

Setting Up a Simple One-Arm Multiple Subnet Topology

You can use a one-arm topology with multiple subnets when the clients and servers reside on the different subnets. For example, consider a NetScaler deployed in one-arm mode for managing servers S1, S2, and S3, with the servers connected to switch SW1 on the network. A vserver of type HTTP is configured on the NetScaler, and HTTP services are running on the servers. These three servers are on the private subnet, so a subnet IP address (SNIP) is configured to communicate with them. The Use Subnet IP address (USNIP) option must be enabled so that the NetScaler uses the SNIP instead of a MIP. As shown in the following figure, the virtual IP address (VIP) is on public subnet 217.60.10.0/24; the NSIP, SNIP, and the server IP addresses are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for One-Arm Mode, Multiple Subnets



Task overview: To deploy a NetScaler in one-arm mode with multiple subnets

1. Configure the NSIP and the default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.

2. Configure the SNIP and enable the USNIP option, as described in the Configuring Subnet IP Addresses (SNIPs).

3. Configure the vserver and the services, as described in Creating a Virtual Server and Configuring Services.

4. Connect one of the network interfaces to the switch.

Configuring System Management Settings

Once your initial configuration is in place, you can configure settings to define the behavior of the Citrix® NetScaler® appliance and facilitate connection management. You have a number of options for handling HTTP requests and responses. Routing, bridging, and MAC based forwarding modes are available for handling packets not addressed to the NetScaler. You can define the characteristics of your network interfaces and can aggregate the interfaces. To prevent timing problems, you can synchronize the NetScaler clock with a Network Time Protocol (NTP) server. The NetScaler can operate in various DNS modes, including as an authoritative domain name server (ADNS). You can set up SNMP for system management and customize syslog logging of system events. Before deployment, verify that your configuration is complete and correct.

Configuring System Settings

Configuration of system settings includes basic tasks such as configuring HTTP ports to enable connection keep-alive and server offload, setting the maximum number of connections for each server, and setting the maximum number of requests per connection. You can enable client IP address insertion for situations in which a proxy IP address is not suitable, and you can change the HTTP cookie version.

You can also configure a NetScaler to open FTP connections on a controlled range of ports instead of ephemeral ports for data connections. This improves security, because opening all ports on the firewall is insecure. You can set the range anywhere from 1,024 to 64,000.

Before deployment, go through the verification checklists to verify your configuration. To configure HTTP parameters and the FTP port range, use the NetScaler configuration utility.

You can modify the types of HTTP parameters described in the following table.

Table 1. HTTP Parameters

Parameter Type	Specifies
HTTP Port Information	The Web server HTTP ports used by your managed servers. If you specify the ports, the NetScaler can perform request switching for any client request that has a destination port matching a specified port. **Note:** If an incoming client request is not destined for a service or a virtual server that is specifically configured on the NetScaler, the destination port in the request must match one of the globally configured HTTP ports. This allows the NetScaler to perform connection keep-alive and server off-load.

Limits	The maximum number of connections to each managed server, and the maximum number of requests sent over each connection. For example, if set **Max Connections** to 500, and the NetScaler is managing three servers, it can open a maximum of 500 connections to each of the three servers. By default, the NetScaler can create an unlimited number of connections to any of the servers it manages. To specify an unlimited number of requests per connection, set **Max Requests** to 0. **Note:** If you are using the Apache HTTP server, you must set **Max Connections** equal to the value of the MaxClients parameter in the Apache httpd.conf file. Setting this parameter is optional for other web servers.
Client IP Insertion	Enable/disable insertion of the client's IP address into the HTTP request header. You can specify a name for the header field in the adjacent text box. When a Web server managed by a NetScaler receives a mapped IP address or a subnet IP address, the server identifies it as the client's IP address. Some applications need the client's IP address for logging purposes or to dynamically determine the content to be served by the web server. You can enable insertion of the actual client IP address into the HTTP header request sent from the client to one, some, or all servers managed by the NetScaler. You can then access the inserted address through a minor modification to the server (using an Apache module, ISAPI interface, or NSAPI interface).
Cookie Version	The HTTP cookie version to use when COOKIEINSERT persistence is configured on a virtual server. The default, version 0, is the most common type on the Internet. Alternatively, you can specify version 1.
Requests/Responses	Options for handling certain types of requests, and enable/disable logging of HTTP error responses.
Server Header Insertion	Insert a server header in NetScaler-generated HTTP responses.

To configure HTTP parameters by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change HTTP parameters**.
3. In the **Configure HTTP parameters** dialog box, specify values for some or all of the parameters that appear under the headings listed in the table above.
4. Click **OK**.

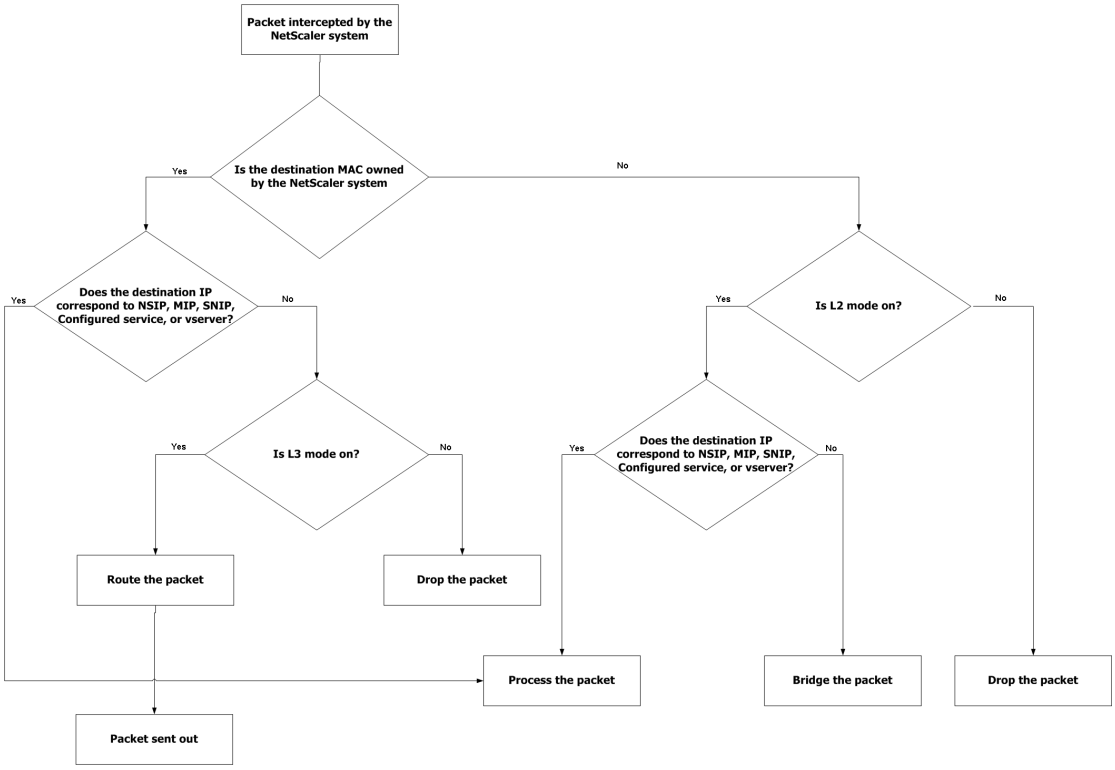
To set the FTP port range by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change global system settings**.
3. Under **FTP Port Range**, in the **Start Port** and **End Port** text boxes, type the lowest and highest port numbers, respectively, for the range you want to specify (for example, 5000 and 6000).
4. Click **OK**.

Configuring Modes of Packet Forwarding

The NetScaler can either route or bridge packets that are not destined for an IP address owned by the NetScaler (that is, the IP address is not the NSIP, a MIP, a SNIP, a configured service, or a configured vserver). By default, L3 mode (routing) is enabled and L2 mode (bridging) is disabled, but you can change the configuration. The following flow chart shows how the NetScaler evaluates packets and either processes, routes, bridges, or drops them.

Figure 1. Interaction between Layer 2 and Layer 3 Modes



A NetScaler can use the following modes to forward the packets it receives:

- Layer 2 (L2) Mode
- Layer 3 (L3) Mode
- MAC-Based Forwarding Mode

Enabling and Disabling Layer 2 Mode

Layer 2 mode controls the Layer 2 forwarding (bridging) function. You can use this mode to configure a NetScaler to behave as a Layer 2 device and bridge the packets that are not destined for it. When this mode is enabled, packets are not forwarded to any of the MAC addresses, because the packets can arrive on any interface of the NetScaler and each interface has its own MAC address.

With Layer 2 mode disabled (which is the default), a NetScaler drops packets that are not destined for one of its MAC address. If another Layer 2 device is installed in parallel with a NetScaler, Layer 2 mode must be disabled to prevent bridging (Layer 2) loops. You can use the configuration utility or the command line to enable Layer 2 mode.

Note: The NetScaler does not support spanning tree protocol. To avoid loops, if you enable L2 mode, do not connect two interfaces on the NetScaler to the same broadcast domain.

To enable or disable Layer 2 mode by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable/disable Layer 2 mode and verify that it has been enabled/disabled:

- `enable ns mode <Mode>`
- `disable ns mode <Mode>`
- `show ns mode`

Examples

```
> enable ns mode l2
 Done
> show ns mode

      Mode                  Acronym          Status
      -------               -------          ------
 1)    Fast Ramp               FR              ON
 2)    Layer 2 mode             L2              ON
 .
 .
 .
 Done
>

> disable ns mode l2
```

```
 Done
> show ns mode

      Mode                  Acronym          Status
      -------               -------          ------
1)    Fast Ramp               FR               ON
2)    Layer 2 mode            L2               OFF
.
.
.
 Done
>
```

To enable or disable Layer 2 mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change modes**.
3. In the **Configure Modes** dialog box, to enable Layer 2 mode, select the **Layer 2 Mode** check box. To disable Layer 2 mode, clear the check box.
4. Click **OK**. The **Enable/Disable Mode(s)?** message appears in the details pane.
5. Click **Yes**.

Enabling and Disabling Layer 3 Mode

Layer 3 mode controls the Layer 3 forwarding function. You can use this mode to configure a NetScaler to look at its routing table and forward packets that are not destined for it. With Layer 3 mode enabled (which is the default), a NetScaler performs route table lookups and forwards all packets that are not destined for any NetScaler-owned IP address. If you disable Layer 3 mode, the NetScaler drops these packets.

To enable or disable Layer 3 mode by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable/disable Layer 3 mode and verify that it has been enabled/disabled:

- `enable ns mode <Mode>`
- `disable ns mode <Mode>`
- `show ns mode`

Examples

```
> enable ns mode l3
 Done
> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
 1)   Fast Ramp                 FR               ON
 2)   Layer 2 mode               L2               OFF
 .
 .
 .
 9)   Layer 3 mode (ip forwarding)   L3               ON
 .
 .
 .
 Done
>

> disable ns mode l3
 Done
> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
```

```
1)   Fast Ramp                  FR              ON
2)   Layer 2 mode                L2              OFF
.
.
.
9)   Layer 3 mode (ip forwarding)  L3              OFF
.
.
.
Done
>
```

To enable or disable Layer 3 mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change modes**.
3. In the **Configure Modes** dialog box, to enable Layer 3 mode, select the **Layer 3 Mode (IP Forwarding)** check box. To disable Layer 3 mode, clear the check box.
4. Click **OK**. The **Enable/Disable Mode(s)?** message appears in the details pane.
5. Click **Yes**.

Enabling and Disabling MAC-Based Forwarding Mode

You can use MAC-based forwarding to process traffic more efficiently and avoid multiple-route or ARP lookups when forwarding packets, because the NetScaler remembers the MAC address of the source. To avoid multiple lookups, the NetScaler caches the source MAC address of every connection for which it performs an ARP lookup, and it returns the data to the same MAC address.

MAC-based forwarding is useful when you use VPN devices because the NetScaler ensures that all traffic flowing through a particular VPN passes through the same VPN device.

The following figure shows the process of MAC-based forwarding.

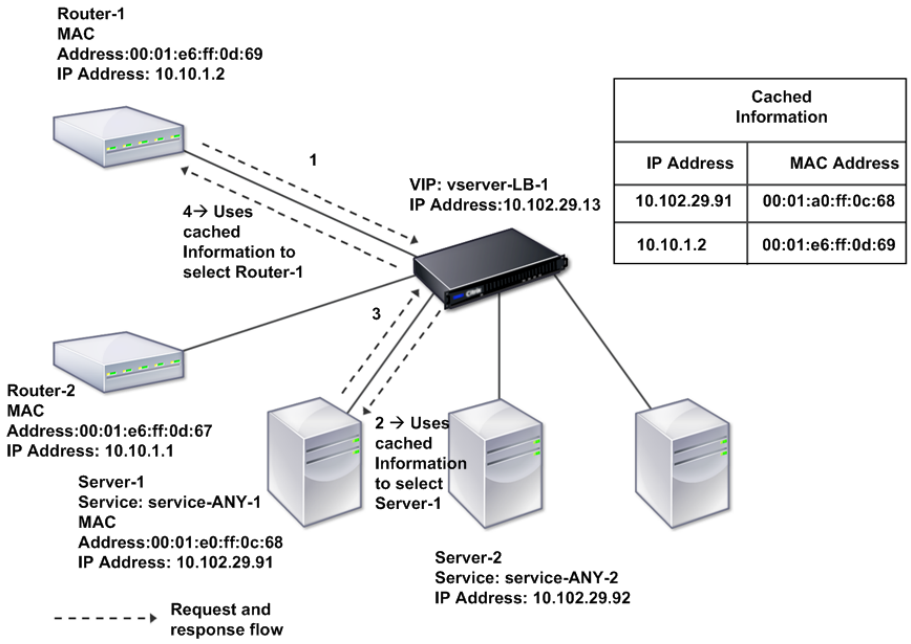


Figure 1. MAC-Based Forwarding Process

When MAC-based forwarding is enabled, a NetScaler caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server responds through a NetScaler, the NetScaler sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when a NetScaler initiates a connection, it uses the route and ARP tables for the lookup function. To enable MAC-based forwarding, use the configuration utility or the command line.

Some deployments require the incoming and outgoing paths to flow through different routers. In these situations, MAC-based forwarding breaks the topology design. For a global server load balancing (GSLB) site that requires the incoming and outgoing paths to flow through different routers, you must disable MAC-based forwarding and use the NetScaler unit's default router as the outgoing router.

With MAC-based forwarding disabled and Layer 2 or Layer 3 connectivity enabled, a route table can specify separate routers for outgoing and incoming connections. To disable MAC-based forwarding, use the configuration utility or the command line.

To enable or disable MAC-based forwarding by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable/disable MAC-based forwarding mode and verify that it has been enabled/disabled:

- `enable ns mode <Mode>`
- `disable ns mode <Mode>`
- `show ns mode`

Example

```
> enable ns mode mbf
 Done
> show ns mode

      Mode                  Acronym          Status
      -------               -------          ------
 1)   Fast Ramp               FR               ON
 2)   Layer 2 mode             L2               OFF
 .
 .
 .
 6)   MAC-based forwarding      MBF              ON
 .
 .
 .
 Done
>

> disable ns mode mbf
 Done
```

```
> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
 1)   Fast Ramp                 FR               ON
 2)   Layer 2 mode               L2               OFF
 .
 .
 .
 6)   MAC-based forwarding       MBF              OFF
 .
 .
 .
 Done
>
```

To enable or disable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features** group, click **Change modes**.
3. In the **Configure Modes** dialog box, to enable MAC-based forwarding mode, select the **MAC Based Forwarding** check box. To disable MAC-based forwarding mode, clear the check box.
4. Click **OK**. The **Enable/Disable Mode(s)?** message appears in the details pane.
5. Click **Yes**.

Configuring Clock Synchronization

You can configure your NetScaler to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network. NTP uses User Datagram Protocol (UDP) port 123 as its transport layer. You have to add NTP servers in the NTP configuration file so that the NetScaler periodically gets updates from these servers.

If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site at http://www.ntp.org/.

To configure clock synchronization on your NetScaler

1. Log on to the NetScaler command line and enter the `shell` command.

2. At the shell prompt, copy the ntp.conf file from the /etc directory to the /nsconfig directory. If the file already exists in the /nsconfig directory, make sure that you remove the following entries from the ntp.conf file:

   ```
   restrict localhost

   restrict 127.0.0.2
   ```

   These entries are required only if you want to run the device as a time server. However, this feature is not supported on the NetScaler.

3. Edit /nsconfig/ntp.conf by typing the IP address for the desired NTP server under the file's server and restrict entries.

4. Create a file named rc.netscaler in the /nsconfig directory, if the file does not already exist in the directory.

5. Edit /nsconfig/rc.netscaler by adding the following entry: `/usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &`

   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   Note: If the time difference between the NetScaler and the time server is more than 1000 sec, the ntpd service terminates with a message to the NetScaler log. To avoid this, you need to start ntpd with the **-g** option,which forcibly syncs the time. Add the following entry in /nsconfig/rc.netscaler:

   ```
   /usr/sbin/ntpd -g -c /nsconfig/ntp.conf -l /var/log/ntpd.log &
   ```

   If you do not want to forcibly sync the time when there is a large difference, you can set the date manually and then start ntpd again. You can check the time difference between the NetScaler and the time server by executing the following command in the shell:

   ntpdate -q <IP address or domain name of the NTP server>

6. Reboot the NetScaler to enable clock synchronization.

   Note: If you want to start time synchronization before you restart the NetScaler, you can enter the

   /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ ntpd.log &

   command (which you added to the rc.netscaler file in step 5) at the shell prompt.

Configuring DNS

You can configure a NetScaler to function as an Authoritative Domain Name Server (ADNS), DNS proxy server, End Resolver, or Forwarder. You can add DNS resource records such as SRV Records, AAAA Records, A Records, MX Records, NS Records, CNAME Records, PTR Records, and SOA Records. Also, the NetScaler can balance the load on external DNS servers.

A common practice is to configure a NetScaler as a forwarder. For this configuration, you need to add external name servers. After you have added the external servers, you should verify that your configuration is correct.

For other configurations, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing vserver as the name server.

When adding name servers, you can specify IP addresses or virtual IP addresses (VIPs). If you use IP addresses, the NetScaler load balances requests to the configured name servers in a round robin manner. If you use VIPs, you can specify any load balancing method. For information about using a VIP, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To add a name server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a name server and verify the configuration:

- `add dns nameServer <IP>`
- `show dns nameServer <IP>`

  Example

```
> add dns nameServer 10.102.29.10
 Done
> show dns nameServer 10.102.29.10
1)      10.102.29.10  -  State: DOWN
 Done
>
```

To add a name server by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Name Servers**.
2. In the details pane, click **Add**.
3. In the **Create Name Server** dialog box, select **IP Address**.
4. In the **IP Address** text box, type the IP address of the name server (for example, `10.102.29.10`). If you are adding an external name server, clear the **Local** check box.
5. Click **Create**, and then click **Close**.
6. Verify that the name server you added appears in the **Name Servers** pane.

Configuring SNMP

The Simple Network Management Protocol (SNMP) network management application, running on a external computer, queries the SNMP agent on the NetScaler. The agent searches the management information base (MIB) for data requested by the network management application and sends the data to the application.

SNMP monitoring uses traps messages and alarms. SNMP traps messages are asynchronous events that the agent generates to signal abnormal conditions, which are indicated by alarms. For example, if you want to be informed when CPU utilization is above 90 percent, you can set up an alarm for that condition. The following figure shows a network with a NetScaler that has SNMP enabled and configured.

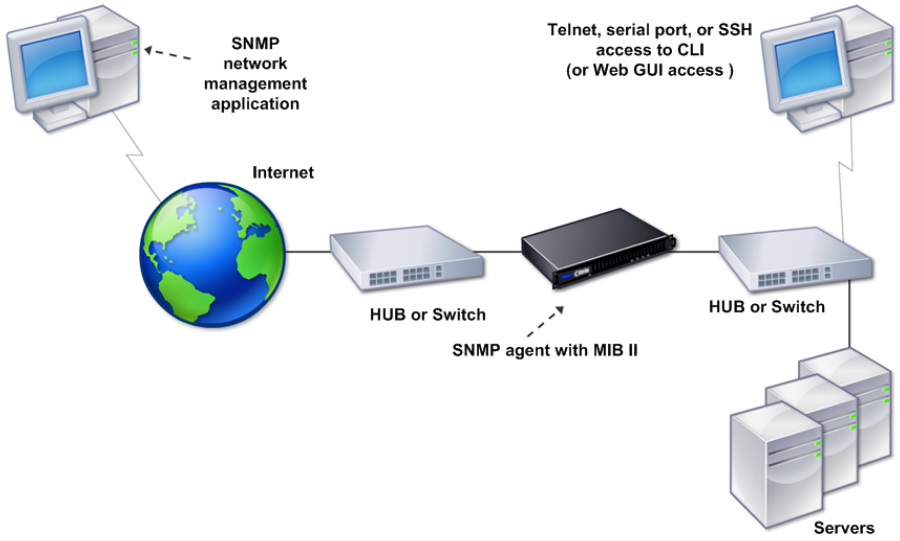


Figure 1. SNMP on the NetScaler

The SNMP agent on a NetScaler supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3). Because it operates in bilingual mode, the agent can handle SNMPv2 queries, such as Get-Bulk, and SNMPv1 queries. The SNMP agent also sends traps compliant with SNMPv2 and supports SNMPv2 data types, such as counter64. SNMPv1 managers (programs on other servers that request SNMP information from the NetScaler) use the NS-MIB-smiv1.mib file when processing SNMP queries. SNMPv2 managers use the NS-MIB-smiv2.mib file.

The NetScaler supports the following enterprise-specific MIBs:

A subset of standard MIB-2 groups

Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.

A system enterprise MIB

Provides system-specific configuration and statistics.

To configure SNMP, you specify which managers can query the SNMP agent, add SNMP trap listeners that will receive the SNMP trap messages, and configure SNMP Alarms.

Adding SNMP Managers

You can configure a workstation running a management application that complies with SNMP version 1, 2, or 3 to access a NetScaler. Such a workstation is called an SNMP manager. If you do not specify an SNMP manager on the NetScaler, the NetScaler accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the NetScaler accepts and responds to SNMP queries from only those specific IP addresses. When specifying the IP address of an SNMP manager, you can use the netmask parameter to grant access from entire subnets. You can add a maximum of 100 SNMP managers or networks.

To add an SNMP manager by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an SNMP manager and verify the configuration:

- `add snmp manager <IPAddress> ... [-netmask <netmask>]`
- `show snmp manager <IPAddress>`

  Example

```
> add snmp manager 10.102.29.5 -netmask 255.255.255.255
 Done
> show snmp manager 10.102.29.5
1)     10.102.29.5       255.255.255.255
 Done
>
```

To add an SNMP manager by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Managers**.
2. In the details pane, click **Add**.
3. In the **Add SNMP Manager** dialog box, in the **IP Address** text box, type the IP address of the workstation running the management application (for example, `10.102.29.5`).
4. Click **Create**, and then click **Close**.
5. Verify that the SNMP manager you added appears in the **Details** section at the bottom of the pane.

Adding SNMP Traps Listeners

After configuring the alarms, you need to specify the trap listener to which the NetScaler will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

To add an SNMP trap listener by using the NetScaler command line

At the NetScaler command prompt, type the following command to add an SNMP trap and verify that it has been added:

- `add snmp trap specific <IP>`
- `show snmp trap`

Example

```
> add snmp trap specific 10.102.29.3
 Done
> show snmp trap
Type       DestinationIP    DestinationPort  Version    SourceIP       Min-Severity   Community
----      -------------   ---------------  -------    --------      ------------  ---------
generic    10.102.29.9     162           V2       NetScaler IP    N/A          public
generic    10.102.29.5     162           V2       NetScaler IP    N/A          public
generic    10.102.120.101   162           V2       NetScaler IP    N/A          public
.
.
.
specific   10.102.29.3     162           V2       NetScaler IP    -           public
 Done
>
```

To add an SNMP trap listener by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Traps**.
2. In the details pane, click **Add**.
3. In the **Add SNMP Trap Destination** dialog box, in the **Destination IP Address** text box, type the IP address (for example, `10.102.29.3`).
4. Click **Create** and then click **Close**.
5. Verify that the SNMP trap you added appears in the **Details** section at the bottom of the pane.

Configuring SNMP Alarms

You configure alarms so that the NetScaler generates a trap message when an event corresponding to one of the alarms occurs. Configuring an alarm consists of enabling the alarm and setting the severity level at which a trap is generated. There are five severity levels: Critical, Major, Minor, Warning, and Informational. A trap is sent only when the severity of the alarm matches the severity specified for the trap.

Some alarms are enabled by default. If you disable an SNMP alarm, the NetScaler will not generate trap messages when corresponding events occur. For example, if you disable the Login-Failure SNMP alarm, the NetScaler will not generate a trap message when a login failure occurs.

To enable or disable an alarm by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable or disable an alarm and verify that it has been enabled or disabled:

- `set snmp alarm <trapName> [-state ENABLED | DISABLED ]`
- `show snmp alarm <trapName>`

Example

```
> set snmp alarm LOGIN-FAILURE -state ENABLED
 Done
> show snmp alarm LOGIN-FAILURE
   Alarm                  Alarm Threshold    Normal Threshold  Time  State     Severity     Logging
   -----                  ---------------    ----------------  ----  --------  ------------ --------
 1) LOGIN-FAILURE              N/A              N/A             N/A   ENABLED   -            ENABLED
 Done
>
```

To set the severity of the alarm by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the severity of the alarm and verify that the severity has been set correctly:

- `set snmp alarm <trapName> [-severity <severity>]`
- `show snmp alarm <trapName>`

Example

```
> set snmp alarm LOGIN-FAILURE -severity Major
 Done
> show snmp alarm LOGIN-FAILURE
   Alarm                    Alarm Threshold    Normal Threshold  Time  State     Severity     Logging
   -----                   ---------------   ----------------  ----  --------  ------------ --------
 1) LOGIN-FAILURE               N/A              N/A             N/A   ENABLED    Major       ENABLED
 Done
>
```

To configure alarms by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Alarms**.
2. In the details pane, select an alarm (for example, **LOGIN-FAILURE**), and then click **Open**.
3. In the **Configure SNMP Alarm** dialog box, to enable the alarm, select the **Enable** check box. To disable the alarm, clear the **Enable** check box.
4. In the **Severity** drop-down list, select a severity option (for example, **Major**).
5. Click **OK**, and then click **Close**.
6. Verify that the parameters for the SNMP alarm you configured are correctly configured by viewing the **Details** section at the bottom of the pane.

Configuring Syslog

You can customize logging of NetScaler and Access Gateway Enterprise Edition access events for the needs of your site. You can direct these logs either to files on the NetScaler or to external log servers. The NetScaler uses the Audit Server Logging feature for logging the states and status information collected by different modules in the kernel and by user-level daemons.

Syslog is used to monitor a NetScaler and to log connections, statistics, and so on. You can customize the two logging functions for system events messaging and syslog. The NetScaler internal event message generator passes log entries to the syslog server. The syslog server accepts these log entries and logs them. For more information about the Audit Server Logging feature, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Verifying the Configuration

After you finish configuring your system, complete the following checklists to verify your configuration.

Configuration Checklist

- The build running is:
- There are no incompatibility issues. (Incompatibility issues are documented in the build's release notes.)
- The port settings (speed, duplex, flow control, monitoring) are the same as the switch's port.
- Enough mapped IP addresses have been configured to support all server-side connections during peak times.
    - The number of configured mapped IP addresses is: ____
    - The expected number of simultaneous server connections is:

      [ ] 62,000 [ ] 124,000 [ ] Other____

Topology Configuration Checklist

The routes have been used to resolve servers on other subnets.

The routes entered are:

_____

- If the NetScaler is in a public-private topology, reverse NAT has been configured.
- The failover (high availability) settings configured on the NetScaler resolve in a one arm or two-arm configuration. All unused network interfaces have been disabled:

  _____
  _____
- If the NetScaler is placed behind an external load balancer, then the load balancing policy on the external load balancer is not "least connection."

  The load balancing policy configured on the external load balancer is:

  _____
- If the NetScaler is placed in front of a firewall, the session time-out on the firewall is set to a value greater than or equal to 300 seconds.

  The value configured for the session time-out is: _____

Server Configuration Checklist

- "Keep-alive" has been enabled on all the servers.

  The value configured for the keep-alive time-out is: _____

- The default gateway has been set to the correct value. (The default gateway should either be a NetScaler or upstream router.) The default gateway is: _____

- The server port settings (speed, duplex, flow control, monitoring) are the same as the switch port settings.

- If the Microsoft® Internet Information Server is used, buffering is enabled on the server.

- If an Apache Server is used, the MaxConn (maximum number of connections) parameter is configured on the server and on the NetScaler.

  The MaxConn (maximum number of connections) value that has been set is: _____

- If a NetScape® Enterprise Server™ is used, the maximum requests per connection parameter is set on the NetScaler.

  The maximum requests per connection value that has been set is: _____

Software Features Configuration Checklist

- Does the Layer 2 mode feature need to be disabled? (Disable if another Layer 2 device is working in para

  Reason for enabling or disabling: _____

- Does the MAC-based forwarding feature need to be disabled? (If the MAC address used by return traffic is

  Reason for enabling or disabling: _____

- Does host-based reuse need to be disabled? (Is there virtual hosting on the servers?)

  Reason for enabling or disabling: _____

- Do the default settings of the surge protection feature need to be changed?

  Reason for changing or not changing: _____

Access Checklist

- The system IPs can be pinged from the client-side network.

- The system IPs can be pinged from the server-side network.
- The managed server(s) can be pinged through the NetScaler.
- Internet hosts can be pinged from the managed servers.
- The managed server(s) can be accessed through the browser.
- The Internet can be accessed from managed server(s) using the browser.
- The system can be accessed using SSH.
- Admin access to all managed server(s) is working.

Note: When you are using the ping utility, ensure that the pinged server has ICMP ECHO enabled, or your ping will not succeed.

Firewall Checklist

The following firewall requirements have been met:

- UDP 161 (SNMP)
- UDP 162 (SNMP trap)
- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

Load Balancing Traffic on a NetScaler

The load balancing feature distributes client requests across multiple servers to optimize resource utilization. In a real-world scenario with a limited number of servers providing service to a large number of clients, a server can become overloaded and degrade the performance of the server farm. A Citrix® NetScaler® appliance uses load balancing criteria to prevent bottlenecks by forwarding each client request to the server best suited to handle the request when it arrives.

To configure load balancing, you define a virtual server (vserver) to proxy multiple servers in a server farm and balance the load among them.

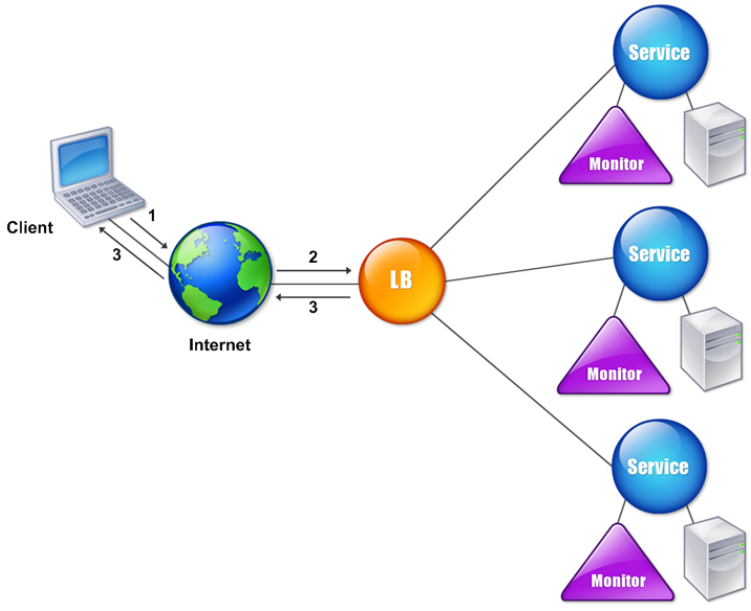
How Load Balancing Works

When a client initiates a connection to the server, a vserver terminates the client connection and initiates a new connection with the selected server, or reuses an existing connection with the server, to perform load balancing. The load balancing feature provides traffic management from Layer 4 (TCP and UDP) through Layer 7 (FTP, HTTP, and HTTPS).

The NetScaler uses a number of algorithms, called load balancing methods, to determine how to distribute the load among the servers. The default load balancing method is the Least Connections method.

A typical load balancing deployment consists of the entities described in the following figure.

Figure 1. Load Balancing Architecture



The entities function as follows:

- **Vserver**. An entity that is represented by an IP address, a port, and a protocol. The vserver IP address (VIP) is usually a public IP address. The client sends connection requests to this IP address. The vserver represents a bank of servers.

- **Service**. A logical representation of a server or an application running on a server. Identifies the server's IP address, a port, and a protocol. The services are bound to the vservers.

- **Server object**. An entity that is represented by an IP address. The server object is created when you create a service. The IP address of the service is taken as the name of the server object. You can also create a server object and then create services by using the server object.

- **Monitor**. An entity that tracks the health of the services. The NetScaler periodically probes the servers using the monitor bound to each service. If a server does not respond within a specified response timeout, and the specified number of probes fails, the service is marked DOWN. The NetScaler then performs load balancing among the remaining services.

Configuring Load Balancing

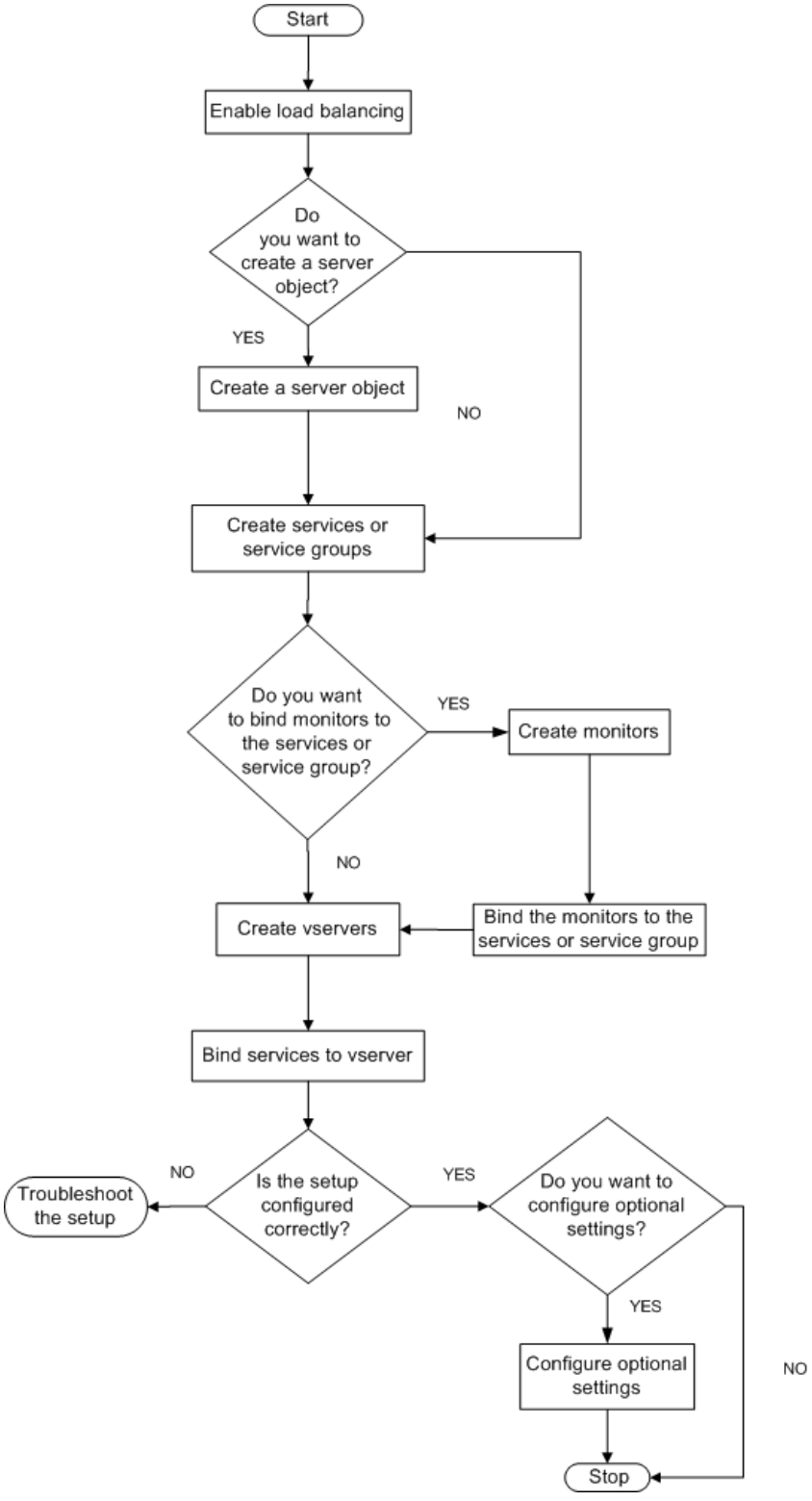
To configure load balancing, you must first create services. Then, you create vservers and bind the services to the vservers. By default, the NetScaler binds a monitor to each service. After binding the services, verify your configuration by making sure that all of the settings are correct.

Note: After you deploy the configuration, you can display statistics that show how the entities in the configuration are performing. Use the statistical utility or the `stat lb vserver <vserverName>` command.

Optionally, you can assign weights to a service. The load balancing method then uses the assigned weight to select a service. For getting started, however, you can limit optional tasks to configuring some basic persistence settings, for sessions that must maintain a connection to a particular server, and some basic configuration-protection settings.

The following flow chart illustrates the sequence of the configuration tasks.

Figure 1. Sequence of Tasks to Configure Load Balancing


Start
Enable load balancing
Do you want to create a server object?
YES
Create a server object
NO
Create services or service groups
Do you want to bind monitors to the services or service group?
YES
Create monitors
NO
Create vservers
Bind the monitors to the services or service group
Bind services to vserver
Is the setup configured correctly?
NO
Troubleshoot the setup
YES
Do you want to configure optional settings?
YES
Configure optional settings
NO
Stop

Enabling Load Balancing

Before configuring load balancing, make sure that the load balancing feature is enabled.

To enable load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable load balancing and verify that it is enabled:

- `enable feature lb`
- `show feature`

Example

```
> enable feature lb
 Done
> show feature

      Feature                Acronym          Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               OFF
3)    Load Balancing           LB               ON
.
.
.
9)    SSL Offloading           SSL              ON
.
.
.
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Load Balancing** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message, click **Yes**.

Configuring Services and a Vserver

When you have identified the services you want to load balance, you can implement your initial load balancing configuration by creating the service objects, creating a load balancing vserver, and binding the service objects to the vserver.

To implement the initial load balancing configuration by using the NetScaler command line

At the NetScaler command prompt, type the following commands to implement and verify the initial configuration:

- `add service <name> <IPaddress> <serviceType> <port>`
- `add lb vserver <vServerName> <serviceType> [<IPaddress> <port>]`
- `bind lb vserver <name> <serviceName>`
- `show service bindings <serviceName>`

  Example

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
 Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
 Done
> bind lb vserver vserver-LB-1 service-HTTP-1
 Done
> show service bindings service-HTTP-1
        service-HTTP-1 (10.102.29.5:80) - State : DOWN

                1)      vserver-LB-1 (10.102.29.60:80) - State : DOWN
 Done
```

To implement the initial load balancing configuration by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the details pane, under **Getting Started**, click **Load Balancing wizard**, and follow the instructions to create a basic load balancing setup.
3. Return to the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
4. Select the vserver that you configured and verify that the parameters displayed at the bottom of the page are correctly configured.
5. Click **Open**.
6. Verify that each service is bound to the vserver by confirming that the **Active** check box is selected for each service on the **Services** tab.

Choosing and Configuring Persistence Settings

You must configure persistence on a vserver if you want to maintain the states of connections on the servers represented by that vserver (for example, connections used in e-commerce). The NetScaler then uses the configured load balancing method for the initial selection of a server, but forwards to that same server all subsequent requests from the same client.

If persistence is configured, it overrides the load balancing methods once the server has been selected. If the configured persistence applies to a service that is down, the NetScaler uses the load balancing methods to select a new service, and the new service becomes persistent for subsequent requests from the client. If the selected service is in an Out Of Service state, it continues to serve the outstanding requests but does not accept new requests or connections. After the shutdown period elapses, the existing connections are closed. The following table lists the types of persistence that you can configure.

Table 1. Limitations on Number of Simultaneous Persistent Connections

Persistence Type	Persistent Connections
Source IP, SSL Session ID, Rule, DESTIP, SRCIPDESTIP	250K
CookieInsert, URL passive, Custom Server ID	Memory limit. In case of CookieInsert, if time out is not 0, any number of connections is allowed until limited by memory.

If the configured persistence cannot be maintained because of a lack of resources on a NetScaler, the load balancing methods are used for server selection. Persistence is maintained for a configured period of time, depending on the persistence type. Some persistence types are specific to certain vservers. The following table shows the relationship.

Table 2. Persistence Types Available for Each Type of Vserver

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL Passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO

SRCIPDESTIP	N/A	N/A	YES	YES	N/A
DESTIP	N/A	N/A	YES	YES	N/A

You can also specify persistence for a group of vservers. When you enable persistence on the group, the client requests are directed to the same selected server regardless of which vserver in the group receives the client request. When the configured time for persistence elapses, any vserver in the group can be selected for incoming client requests.

Two commonly used persistence types are persistence based on cookies and persistence based on server IDs in URLs. For more information about all persistence types, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Configuring Persistence Based on Cookies

When you enable persistence based on cookies, the NetScaler adds an HTTP cookie into the Set-Cookie header field of the HTTP response. The cookie contains information about the service to which the HTTP requests must be sent. The client stores the cookie and includes it in all subsequent requests, and the NetScaler uses it to select the service for those requests. You can use this type of persistence on vservers of type HTTP or HTTPS.

The NetScaler inserts the cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

where:

- <NSC_XXXX> is the vserver ID that is derived from the vserver name.
- <ServiceIP> is the hexadecimal value of the IP address of the service.
- <ServicePort> is the hexadecimal value of the port of the service.

The NetScaler encrypts ServiceIP and ServicePort when it inserts a cookie, and decrypts them when it receives a cookie.

Note: If the client is not allowed to store the HTTP cookie, the subsequent requests do not have the HTTP cookie, and persistence is not honored.

By default, the NetScaler sends HTTP cookie version 0, in compliance with the Netscape specification. It can also send version 1, in compliance with RFC 2109.

You can configure a timeout value for persistence that is based on HTTP cookies. Note the following:

- If HTTP cookie version 0 is used, the NetScaler inserts the absolute Coordinated Universal Time (GMT) of the cookie's expiration (the expires attribute of the HTTP cookie), calculated as the sum of the current GMT time on a NetScaler, and the timeout value.
- If an HTTP cookie version 1 is used, the NetScaler inserts a relative expiration time (Max-Age attribute of the HTTP cookie). In this case, the client software calculates the actual expiration time.

Note: Most client software currently installed (Microsoft Internet Explorer and Netscape browsers) understand HTTP cookie version 0; however, some HTTP proxies understand HTTP cookie version 1.

If you set the timeout value to 0, the NetScaler does not specify the expiration time, regardless of the HTTP cookie version used. The expiration time then depends on the client software, and such cookies are not valid if that software is shut down. This persistence type

does not consume any system resources. Therefore, it can accommodate an unlimited number of persistent clients.

An administrator can use the procedure in the following table to change the HTTP cookie version.

To change the HTTP cookie version by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, click **Change HTTP Parameters**.
3. In the **Configure HTTP Parameters** dialog box, under **Cookie**, select **Version 0** or **Version 1**.

Note: For information about the parameters, see the "Load Balancing" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To configure persistence based on cookies by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure persistence based on cookies and verify the configuration:

- `set lb vserver <name> -persistenceType COOKIEINSERT`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        .
        .
        .
        Persistence: COOKIEINSERT (version 0)   Persistence Timeout: 2 min
        .
        .
        .
 Done
>
```

To configure persistence based on cookies by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure persistence (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, in the **Persistence** list, select **COOKIEINSERT**.
4. In the **Time-out (min)** text box, type the time-out value (for example, 2).
5. Click **OK**.
6. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the **Details** section at the bottom of the pane.

Configuring Persistence Based on Server IDs in URLs

The NetScaler can maintain persistence based on the server IDs in the URLs. In a technique called URL passive persistence, the NetScaler extracts the server ID from the server response and embeds it in the URL query of the client request. The server ID is an IP address and port specified as a hexadecimal number. The NetScaler extracts the server ID from subsequent client requests and uses it to select the server.

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. For more information about expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: If the server ID cannot be extracted from the client requests, server selection is based on the load balancing method.

Example: Payload Expression

The expression, URLQUERY contains sid= configures the system to extract the server ID from the URL query of a client request, after matching token sid=. Thus, a request with the URL http://www.citrix.com/index.asp?&sid;=c0a864100050 is directed to the server with the IP address10.102.29.10 and port 80.

The timeout value does not affect this type of persistence, which is maintained as long as the server ID can be extracted from the client requests. This persistence type does not consume any system resources, so it can accommodate an unlimited number of persistent clients.

Note: For information about the parameters, see the “Load Balancing” chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To configure persistence based on server IDs in URLs by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure persistence based on server IDs in URLs and verify the configuration:

- `set lb vserver <name> -persistenceType URLPASSIVE`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        .
        .
        .
        Persistence: URLPASSIVE Persistence Timeout: 2 min
        .
        .
        .
 Done
>
```

To configure persistence based on server IDs in URLs by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure persistence (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, in the **Persistence** list, select **URLPASSIVE**.
4. In the **Time-out (min)** text box, type the time-out value (for example, 2).
5. In the **Rule** text box, enter a valid expression. Alternatively, click **Configure** next to the **Rule** text box and use the **Create Expression** dialog box to create an expression.
6. Click **OK**.
7. Verify that the vserver for which you configured persistence is correctly configured by selecting the vserver and viewing the **Details** section at the bottom of the pane.

Configuring Features to Protect the Load Balancing Configuration

You can configure URL redirection to provide notifications of vserver malfunctions, and you can configure backup vservers to take over if a primary vserver becomes unavailable.

Configuring URL Redirection

You can configure a redirect URL to communicate the status of the NetScaler in the event that a vserver of type HTTP or HTTPS is down or disabled. This URL can be a local or remote link. The NetScaler uses HTTP 302 redirect.

Redirects can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Note: If a load balancing vserver is configured with both a backup vserver and a redirect URL, the backup vserver takes precedence over the redirect URL. In this case, a redirect is used when both the primary and backup vservers are down.

To configure a vserver to redirect client requests to a URL by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a vserver to redirect client requests to a URL and verify the configuration:

- `set lb vserver <name> -redirectURL <URL>`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
        .
        .
        .
        Redirect URL: http://www.newdomain.com/mysite/maintenance
        .
        .
        .
 Done
>
```

To configure a vserver to redirect client requests to a URL by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure URL redirection (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Advanced** tab, in the **Redirect URL** text box, type the URL (for example, `http://www.newdomain.com/mysite/maintenance`), and then click **OK**.
4. Verify that the redirect URL you configured for the server appears in the **Details** section at the bottom of the pane.

Configuring Backup Vservers

If the primary vserver is down or disabled, the NetScaler can direct the connections or client requests to a backup vserver that forwards the client traffic to the services. The NetScaler can also send a notification message to the client regarding the site outage or maintenance. The backup vserver is a proxy and is transparent to the client.

You can configure a backup vserver when you create a vserver or when you change the optional parameters of an existing vserver. You can also configure a backup vserver for an existing backup vserver, thus creating cascaded backup vservers. The maximum depth of cascading backup vservers is 10. The NetScaler searches for a backup vserver that is up and accesses that vserver to deliver the content.

You can configure URL redirection on the primary for use when the primary and the backup vservers are down or have reached their thresholds for handling requests.

Note: If no backup vserver exists, an error message appears, unless the vserver is configured with a redirect URL. If both a backup vserver and a redirect URL are configured, the backup vserver takes precedence.

To configure a backup vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a backup server and verify the configuration:

- `set lb vserver <name> [-backupVserver <string>]`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
        .
        .
        .
        Backup: vserver-LB-2
        .
        .
        .
 Done
>
```

To set up a backup vserver by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure the backup vserver (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Advanced** tab, in the **Backup Virtual Server** list, select the backup vserver (for example, **vserver-LB-2**, and then click **OK**.
4. Verify that the backup vserver you configured appears in the **Details** section at the bottom of the pane.

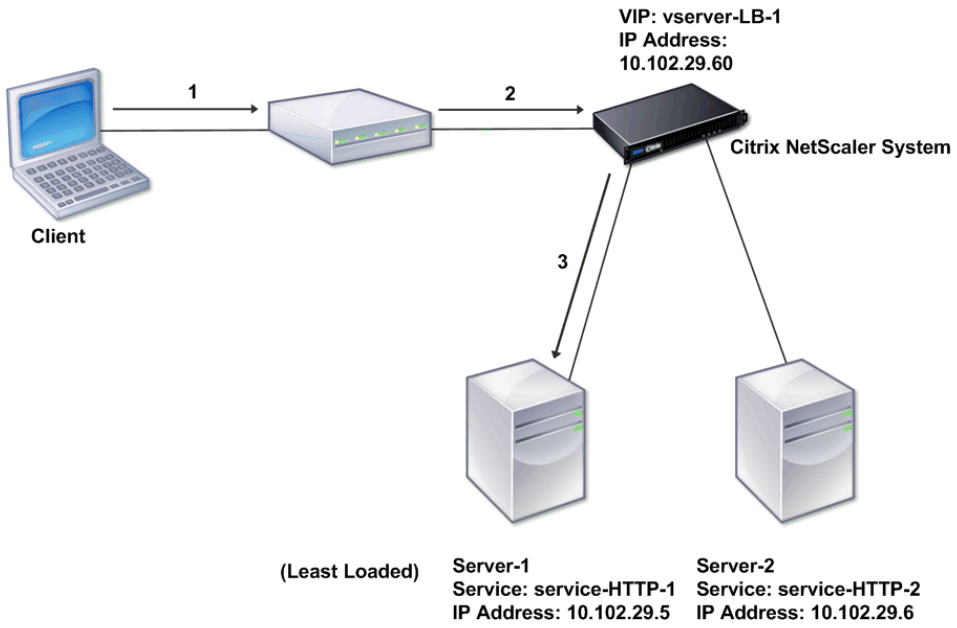
   Note: If the primary server goes down and then comes back up, and you want the backup vserver to function as the primary server until you explicitly reestablish the primary virtual server, select the **Disable Primary When Down** check box.

A Typical Load Balancing Scenario

In a load balancing setup, the NetScalers are logically located between the client and the server farm, and they manage traffic flow to the servers.

The following figure shows the topology of a basic load balancing configuration.

Figure 1. Basic Load Balancing Topology



The vserver selects the service and assigns it to serve client requests. Consider the scenario in the preceding figure, where the services service-HTTP-1 and service-HTTP-2 are created and bound to the vserver named vserver-LB-1. Vserver-LB-1 forwards the client request to either service-HTTP-1 or service-HTTP-2. The system selects the service for each request by using the Least Connections load balancing method. The following table lists the names and values of the basic entities that must be configured on the system.

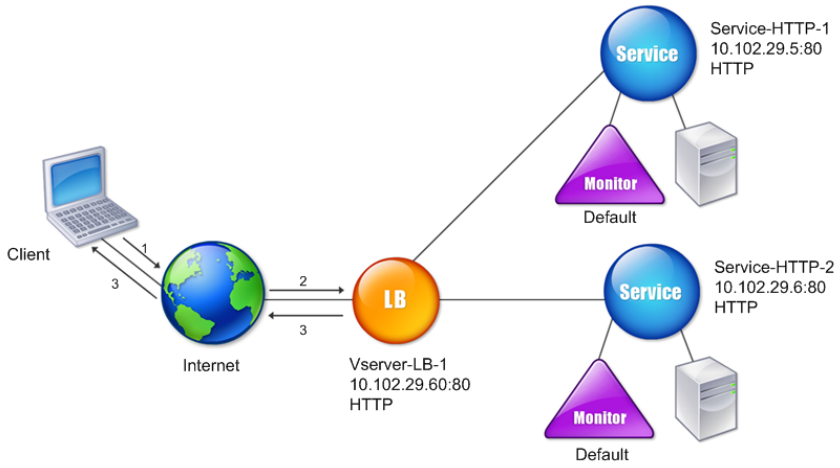
Table 1. LB Configuration Parameter Values

Entity Type	Required parameters and sample values			
	Name	IP Address	Port	Protocol
Vserver	vserver-LB-1	10.102.29.60	80	HTTP
Services	service-HTTP-1	10.102.29.5	8083	HTTP
	service-HTTP-2	10.102.29.6	80	HTTP

Monitors	Default	None	None	None

The following figure shows the load balancing sample values and required parameters that are described in the preceding table.

Figure 2. Load Balancing Entity Model



The following tables list the commands used to configure this load balancing setup by using the NetScaler command line.

Table 2. Initial Configuration Tasks

Task	Command
To enable load balancing	enable feature lb
To create a service named service-HTTP-1	add service service-HTTP-1 10.102.29.5 HTTP 80
To create a service named service-HTTP-2	add service service-HTTP-2 10.102.29.6 HTTP 80
To create a vserver named vserver-LB-1	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
To bind a service named service-HTTP-1 to a vserver named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-1
To bind a service named service-HTTP-2 to a vserver named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-2

For more information about the initial configuration tasks, see Enabling Load Balancing and Configuring Services and a Vserver.

Table 3. Verification Tasks

Task	Command
To view the properties of a vserver named vserver-LB-1	show lb vserver vserver-LB-1
To view the statistics of a vserver named vserver-LB-1	stat lb vserver vserver-LB-1
To view the properties of a service named service-HTTP-1	show service service-HTTP-1
To view the statistics of a service named service-HTTP-1	stat service service-HTTP-1
To view the bindings of a service named service-HTTP-1	show service bindings service-HTTP-1

Table 4. Customization Tasks

Task	Command
To configure persistence on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
To configure COOKIEINSERT persistence on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
To configure URLPassive persistence on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
To configure a vserver to redirect the client request to a URL on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
To set a backup vserver on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

For more information about configuring persistence, see Choosing and Configuring Persistence Settings. For information about configuring a vserver to redirect a client request to a URL and setting up a backup vserver, see Configuring Features to Protect the Load Balancing Configuration.

Accelerating Load Balanced Traffic by Using Compression

Compression is a popular means of optimizing bandwidth usage, and most web browsers support compressed data. If you enable the compression feature, the Citrix® NetScaler® intercepts requests from clients and determines whether the client can accept compressed content. After receiving the HTTP response from the server, the NetScaler examines the content to determine whether it is compressible. If the content is compressible, the NetScaler compresses it, modifies the response header to indicate the type of compression performed, and forwards the compressed content to the client.

NetScaler compression is a policy-based feature. A policy filters requests and responses to identify responses to be compressed, and specifies the type of compression to apply to each response. The NetScaler provides several built-in policies to compress common MIME types such as text/html, text/plain, text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint. You can also create custom policies. The NetScaler does not compress compressed MIME types such as application/octet-stream, binary, bytes, and compressed image formats such as GIF and JPEG.

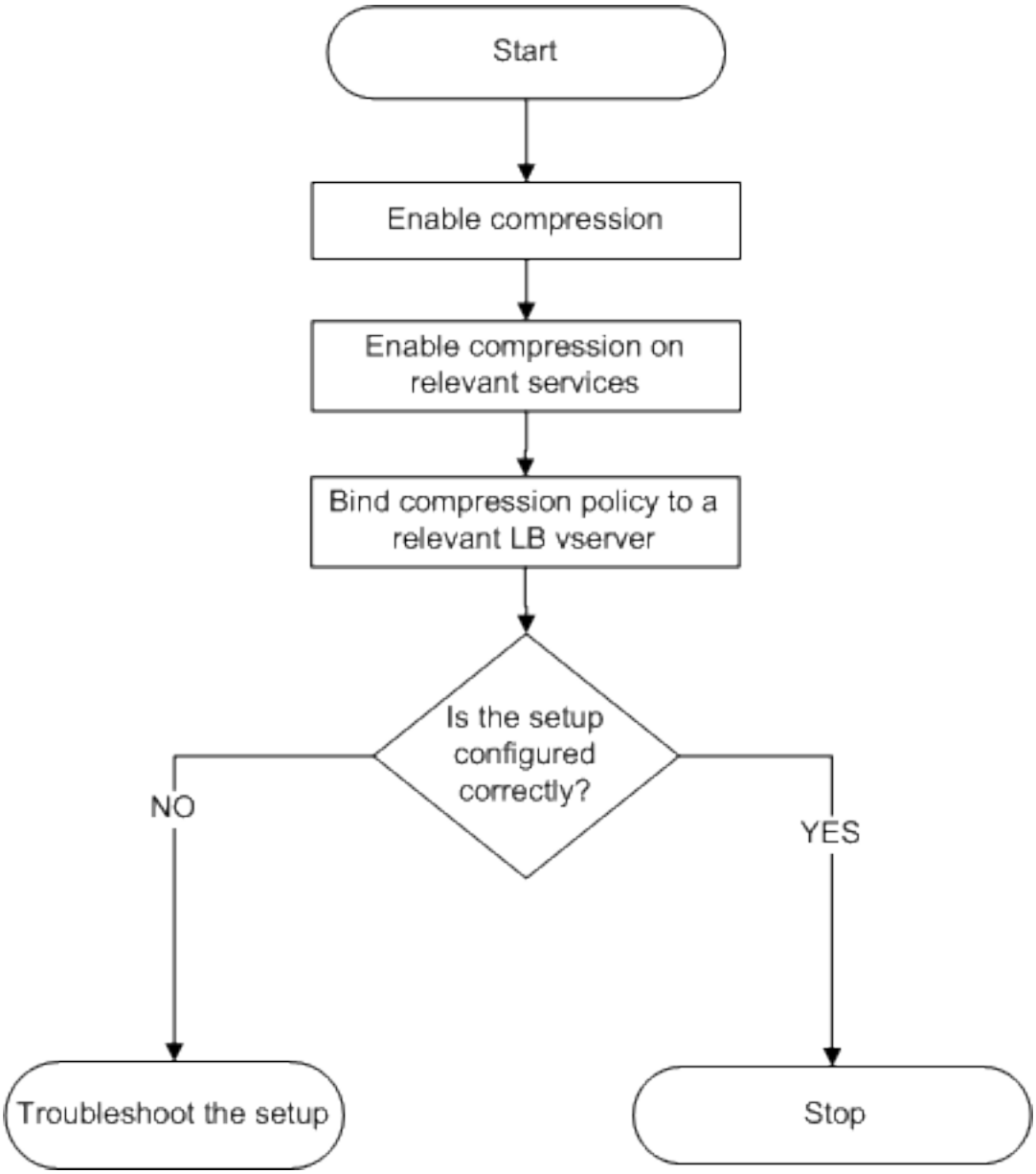
To configure compression, you must enable it globally and on each service that will provide responses that you want compressed. If you have configured vservers for load balancing or content switching, you should bind the polices to the vservers. Otherwise, the policies apply to all traffic that passes through the NetScaler.

Note: For more information about compression, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Compression Configuration Task Sequence

The following flow chart shows the sequence of tasks for configuring basic compression in a load balancing setup.

Figure 1. Sequence of Tasks to Configure Compression



Note: The steps in the above figure assume that load balancing has already been configured. For information about configuring load balancing, or for more information about services, see the *Citrix NetScaler Traffic Management Guide* at

http://support.citrix.com/article/CTX128670.

If you want to configure something other than a basic compression setup, (for example, if you need to configure optional parameters in addition to the required parameters) see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Enabling Compression

By default, compression is not enabled. You must enable the compression feature to allow compression of HTTP responses that are sent to the client.

To enable compression by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable compression and verify the configuration:

- `enable ns feature CMP`
- `show ns feature`

Example

```
> enable ns feature CMP
Done
> show ns feature

Feature                  Acronym          Status
-------                  -------          ------
1)    Web Logging              WL              ON
2)    Surge Protection         SP              OFF
.
7)    Compression Control      CMP             ON
8)    Priority Queuing         PQ              OFF
.
Done
```

To enable compression by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Compression** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Configuring Services to Compress Data

In addition to enabling compression globally, you must enable it on each service that will deliver files to be compressed. To create a service, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To enable compression on a service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable compression on a service and verify the configuration:

- `set service <name> -CMP YES`
- `show service <name>`

Example

```
> show service SVC_HTTP1
SVC_HTTP1 (10.102.29.18:80) - HTTP
State: UP
Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)
Time since last state change: 0 days, 03:03:37.200
Server Name: 10.102.29.18
Server ID : 0   Monitor Threshold : 0
Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
Use Source IP: NO
Client Keepalive(CKA): NO
Access Down Service: NO
TCP Buffering(TCPB): NO
HTTP Compression(CMP): YES
Idle timeout: Client: 180 sec   Server: 360 sec
Client IP: DISABLED
Cacheable: NO
SC: OFF
SP: OFF
Down state flush: ENABLED

1)      Monitor Name: tcp-default
State: DOWN     Weight: 1
Probes: 1095    Failed [Total: 1095 Current: 1095]
Last response: Failure - TCP syn sent, reset received.
Response Time: N/A
Done
```

To enable compression on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure compression (for example, **service-HTTP-1**), and then click **Open**.
3. On the **Advanced** tab, under **Settings**, select the **Compression** check box, and then click **OK.**
4. Verify that, when the service is selected, `HTTP Compression(CMP): ON` appears in the **Details** section at the bottom of the pane.

Binding a Compression Policy to a Virtual Server

If you bind a policy to a virtual server, the policy is evaluated only by the services associated with that virtual server. You can bind compression policies to a virtual server either from the **Configure Virtual Server (Load Balancing)** dialog box or from the **Compression Policy Manager** dialog box. This topic includes instructions to bind compression policies to a load balancing virtual server by using the **Configure Virtual Server (Load Balancing)** dialog box. For information about how you can bind a compression policy to a load balancing virtual server by using the **Compression Policy Manager** dialog box, see Configuring and Binding Policies with the Policy Manager. the "Configuring Advanced Policies" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To bind or unbind a compression policy to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind or unbind a compression policy to a load balancing virtual server and verify the configuration:

- `(bind|unbind) lb vserver <name> -policyName <string>`
- `show lb vserver <name>`

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> show lb vserver lbvip
lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Thu May 28 05:37:21 2009 (+685 ms)
Time since last state change: 19 days, 04:26:50.470
Effective State: UP
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Port Rewrite : DISABLED
No. of Bound Services :  1 (Total)       1 (Active)
Configured Method: LEASTCONNECTION
Current Method: Round Robin, Reason: Bound service's state changed to UP
Mode: IP
Persistence: NONE
```

```
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule:

Bound Service Groups:
1)      Group Name: Service-Group-1

1) Service-Group-1 (10.102.29.252: 80) - HTTP State: UP Weight:

1)      Policy : ns_cmp_msapp Priority:0
Done
```

To bind or unbind a compression policy to a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind or unbind a compression policy (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Policies** tab, click **Compression**.
4. Do one of the following:
   - To bind a compression policy, click **Insert Policy**, and then select the policy you want to bind to the virtual server.
   - To unbind a compression policy, click the name of the policy you want to unbind from the virtual server, and then click **Unbind Policy**.
5. Click **OK.**

Securing Load Balanced Traffic by Using SSL

The Citrix® NetScaler® SSL offload feature transparently improves the performance of web sites that conduct SSL transactions. By offloading CPU-intensive SSL encryption and decryption tasks from the local web server to the NetScaler, SSL offloading ensures secure delivery of web applications without the performance penalty incurred when the server processes the SSL data. Once the SSL traffic is decrypted, it can be processed by all standard services. The SSL protocol works seamlessly with various types of HTTP and TCP data and provides a secure channel for transactions using such data.

To configure SSL, you must first enable it. Then, you configure HTTP or TCP services and an SSL virtual server on the NetScaler, and bind the services to the vserver. You must also add a certificate-key pair and bind it to the SSL virtual server. If you use Outlook Web Access servers, you must create an action to enable SSL support and a policy to apply the action. An SSL virtual server intercepts incoming encrypted traffic and decrypts it by using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the NetScaler for appropriate processing.

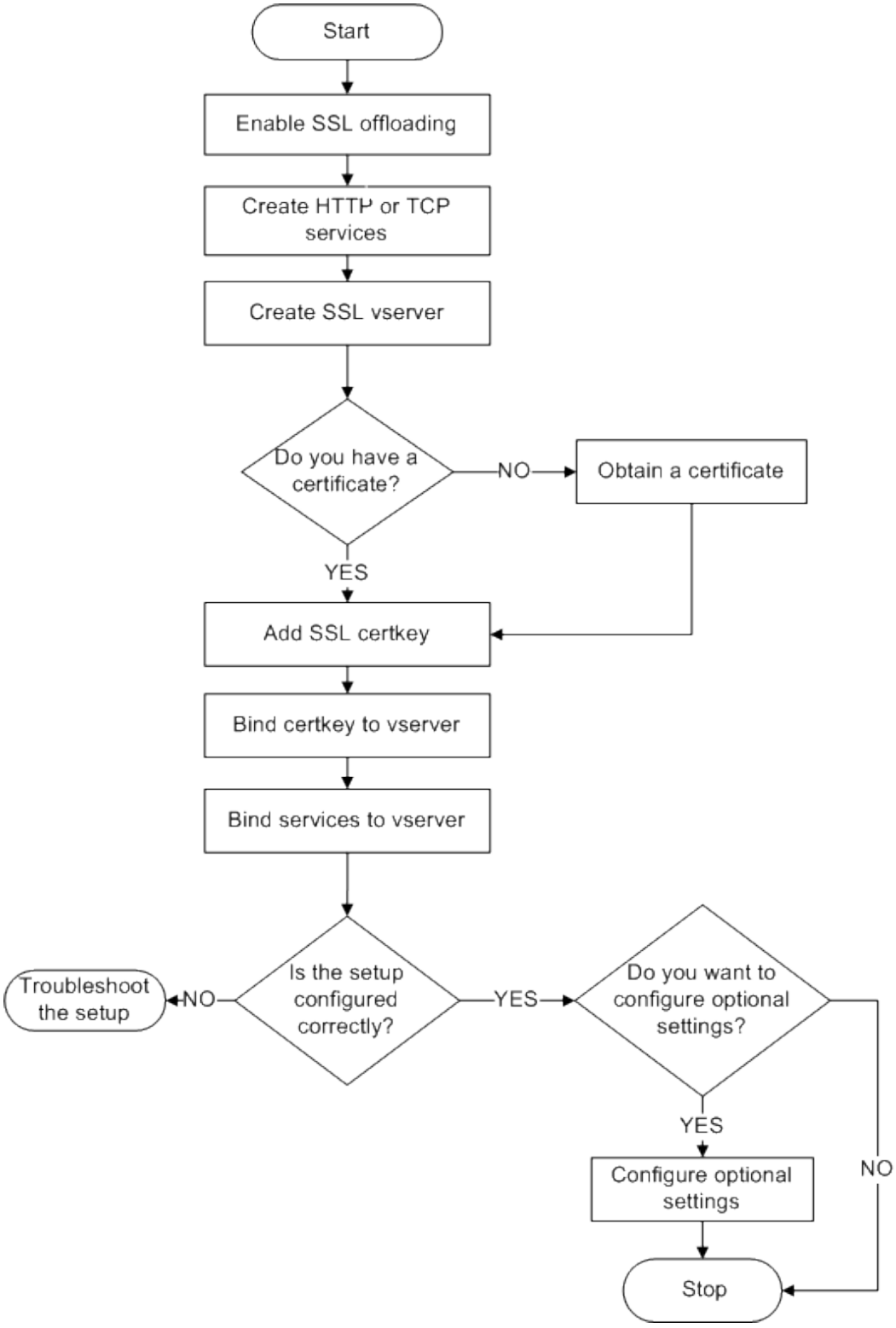
SSL Configuration Task Sequence

To configure SSL, you must first enable it. Then, you must create an SSL virtual server and HTTP or TCP services on the NetScaler. Finally, you must bind a valid SSL certificate and the configured services to the SSL virtual server.

An SSL virtual server intercepts incoming encrypted traffic and decrypts it using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the NetScaler for appropriate processing.

The following flow chart shows the sequence of tasks for configuring a basic SSL offload setup.

Figure 1. Sequence of Tasks to Configure SSL Offloading


Start
Enable SSL offloading
Create HTTP or TCP services
Create SSL vserver
Do you have a certificate?
NO
Obtain a certificate
YES
Add SSL certkey
Bind certkey to vserver
Bind services to vserver
Troubleshoot the setup
NO
Is the setup configured correctly?
YES
Do you want to configure optional settings?
YES
Configure optional settings
NO
Stop

Enabling SSL Offload

You should enable the SSL feature before configuring SSL offload. You can configure SSL-based entities on the system without enabling the SSL feature, but they will not work until you enable SSL.

To enable SSL by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable SSL Offload and and verify the configuration:

- `enable ns feature SSL`
- `show ns feature`

Example

```
> enable ns feature ssl
Done
> show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL ON
2) SurgeProtection SP OFF
3) Load Balancing LB ON . . .
 9) SSL Offloading SSL ON
10) Global Server Load Balancing GSLB ON . .
Done >
```

To enable SSL by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. Select the **SSL Offloading** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Creating HTTP Services

A service on the NetScaler represents an application on a server. Once configured, services are in the disabled state until the NetScaler can reach the server on the network and monitor its status. This topic covers the steps to create an HTTP service.

Note: For TCP traffic, perform the procedures in this and the following topics, but create TCP services instead of HTTP services.

To add an HTTP service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a HTTP service and verify the configuration:

- `add service <name> (<IP> | <serverName>) <serviceType> <port>`
- `show service <name>`

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80
 Done
> show service SVC_HTTP1
        SVC_HTTP1 (10.102.29.18:80) - HTTP
        State: UP
        Last state change was at Wed Jul 15 06:13:05 2009
        Time since last state change: 0 days, 00:00:15.350
        Server Name: 10.102.29.18
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: tcp-default
              State: UP       Weight: 1
              Probes: 4       Failed [Total: 0 Current: 0]
              Last response: Success - TCP syn+ack received.
              Response Time: N/A
```

Done

To add an HTTP service by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Services**.
2. In details pane, click **Add**.
3. In the **Create Service** dialog box, in the **Service Name**, **Server**, and **Port** text boxes, type the name of the service, IP address, and port (for example, `SVC_HTTP1`, `10.102.29.18`, and `80`).
4. In the **Protocol** list, select the type of the service (for example, `HTTP`).
5. Click **Create**, and then click **Close**. The HTTP service you configured appears in the **Services** page.
6. Verify that the parameters you configured are correctly configured by selecting the service and viewing the **Details** section at the bottom of the pane.

For more information about services, see the “Load Balancing” chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Adding an SSL-Based Vserver

In a basic SSL offloading setup, the SSL virtual server intercepts encrypted traffic, decrypts it, and sends the clear text messages to the services that are bound to the virtual server. Offloading CPU-intensive SSL processing to the NetScaler allows the back-end servers to process a greater number of requests.

To add an SSL-based vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL-based vserver and verify the configuration:

- `add lb vserver <name> <serviceType> [<IPAddress> <port>]`
- `show lb vserver <name>`

Example

```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
Done
> show lb vserver vserver-SSL-1
vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS
State: DOWN[Certkey not bound] Last state change was at Tue Jun 16 06:33:08 2009 (
Time since last state change: 0 days, 00:03:44.120
Effective State: DOWN Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
No. of Bound Services : 0 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule: Done
```

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based vserver before you enable it.

To add an SSL-based vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (SSL Offload)** dialog box, in the **Name**, **IP Address**, and **Port** text boxes, type the name of the vserver, IP address, and port (for example, `Vserver-SSL-1`, `10.102.29.50`, and `443`).
4. In the **Protocol** list, select the type of the vserver, for example, **SSL.**
5. Click **Create**, and then click **Close**.
6. Verify that the parameters you configured are correctly configured by selecting the vserver and viewing the **Details** section at the bottom of the pane. The vserver is marked as DOWN because a certificate-key pair and services have not been bound to it.

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based vserver before you enable it.

Binding Services to the SSL Vserver

After decrypting the incoming data, the SSL vserver forwards the data to the services that you have bound to the vserver.

Data transfer between the NetScaler and the servers can be encrypted or in clear text. If the data transfer between the NetScaler and the servers is encrypted, the entire transaction is secure from end to end. For more information about configuring the system for end-to-end security, see the "Secure Socket Layer (SSL) Acceleration" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To bind a service to a vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind service to the SSL vserver and verify the configuration:

- `bind lb vserver <name> <serviceName>`
- `show lb vserver <name>`

Example

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1
Done
> show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) - SSL Type:
ADDRESS State: DOWN[Certkey not bound]
Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)
Time since last state change: 0 days, 00:31:53.70
Effective State: DOWN Client Idle
Timeout: 180 sec
Down state flush: ENABLED Disable Primary Vserver On Down :
DISABLED No. of Bound Services : 1 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver IP and
Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Ru

1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
State: DOWN Weight: 1
Done
```

To bind a service to a vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, select a vserver, and then click **Open**.
3. On the **Services** tab, in the **Active** column, select the check boxes next to the services that you want to bind to the selected vserver.
4. Click **OK.**
5. Verify that the **Number of Bound Services** counter in the **Details** section at the bottom of the pane is incremented by the number of services that you bound to the vserver.

Adding a Certificate Key Pair

An SSL certificate is an integral element of the SSL Key-Exchange and encryption/decryption process. The certificate is used during SSL handshake to establish the identity of the SSL server. You can use a valid, existing SSL certificate that you have on the NetScaler, or you can create your own SSL certificate. The NetScaler supports RSA/DSA certificates of up to 4096 bits.

Note: Citrix recommends that you use a valid SSL certificate that has been issued by a trusted certificate authority. Invalid certificates and self-created certificates are not compatible with all SSL clients.

Before a certificate can be used for SSL processing, you must pair it with its corresponding key. The certificate key pair is then bound to the vserver and used for SSL processing.

To add a certificate key pair by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a certificate key pair and verify the configuration:

- `add ssl certKey <certkeyName> -cert <string> [-key <string>]`
- `show sslcertkey <name>`

Example

```
> add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
 Done
> show sslcertkey CertKey-SSL-1
      Name: CertKey-SSL-1 Status: Valid,
      Days to expiration:4811 Version: 3
      Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer: C=US,ST=California,L=San
      Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
      Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17 21:26:47 2022 GMT
      Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,CN=d efault Public Key Algori
      size: 1024
 Done
```

To add a certificate key pair by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. In the details pane, click **Add**.
3. In the **Install Certificate** dialog box, in the **Certificate-Key Pair Name** text box, type a name for the certificate key pair you want to add, for example, `Certkey-SSL-1`.
4. Under **Details**, in **Certificate File Name**, click **Browse (Appliance)** to locate the certificate. Both the certificate and the key are stored in the /nsconfig/ssl/ folder on the appliance. To use a certificate present on the local system, select **Local**.
5. Select the certificate you want to use, and then click **Select**.
6. In **Private Key File Name**, click **Browse (Appliance)** to locate the private key file. To use a private key present on the local system, select **Local**.
7. Select the key you want to use and click **Select.** To encrypt the key used in the certificate key pair, type the password to be used for encryption in the **Password** text box.
8. Click **Install**.
9. Double-click the certificate key pair and, in the **Certificate Details** window, verify that the parameters have been configured correctly and saved.

Binding an SSL Certificate Key Pair to the Vserver

After you have paired an SSL certificate with its corresponding key, you must bind the certificate key pair to the SSL vserver so that it can be used for SSL processing. Secure sessions require establishing a connection between the client computer and an SSL-based virtual server on the NetScaler. SSL processing is then carried out on the incoming traffic at the virtual server. Therefore, before enabling the SSL virtual server on the NetScaler, you need to bind a valid SSL certificate to the SSL virtual server.

To bind an SSL certificate key pair to a vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an SSL certificate key pair to a vserver and verify the configuration:

- `bind ssl vserver <vServerName> -certkeyName <string>`
- show ssl vserver <name>

  Example

```
 > bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1
Done
> show ssl vserver Vserver-SSL-1

    Advanced SSL configuration for VServer Vserver-SSL-1:
    DH: DISABLED
    Ephemeral RSA: ENABLED Refresh Count: 0
    Session Reuse: ENABLED Timeout: 120 seconds
    Cipher Redirect: ENABLED
    SSLv2 Redirect: ENABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: CertKey-SSL-1 Server Certificate
1) Cipher Name: DEFAULT
   Description: Predefined Cipher Alias
Done
```

To bind an SSL certificate key pair to a vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the vserver to which you want to bind the certificate key pair, for example, **Vserver-SSL-1**, and click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, on the **SSL Settings** tab, under **Available**, select the certificate key pair that you want to bind to the vserver (for example, **Certkey-SSL-1**), and then click **Add**.
4. Click **OK.**
5. Verify that the certificate key pair that you selected appears in the **Configured** area.

Configuring Support for Outlook Web Access

If you use Outlook Web Access (OWA) servers on your NetScaler, you must configure the NetScaler to insert a special header field, FRONT-END-HTTPS: ON, in HTTP requests directed to the OWA servers, so that the servers generate URL links as https:// instead of http://.

Note: You can enable OWA support for HTTP-based SSL vservers and services only. You cannot apply it for TCP-based SSL vservers and services.

To configure OWA support, do the following:

- Create an SSL action to enable OWA support.
- Create an SSL policy.
- Bind the policy to the SSL vserver.

Creating an SSL Action to Enable OWA Support

Before you can enable OWA support, you must creat an SSL action. SSL actions are bound to SSL policies and triggered when incoming data matches the rule specified by the policy.

To create an SSL action to enable OWA support by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

- `add ssl action <name> -OWASupport ENABLED`
- `show SSL action <name>`

```
 > add ssl action Action-SSL-OWA -OWASupport enabled
Done
> show SSL action Action-SSL-OWA
  Name: Action-SSL-OWA
  Data Insertion Action: OWA
  Support: ENABLED
Done
```

To create an SSL action to enable OWA support by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. In the details pane, on the **Actions** tab, click **Add**.
3. In the **Create SSL Action** dialog box, in the **Name** text box, type `Action-SSL-OWA`.
4. Under **Outlook Web Access**, select **Enabled**.
5. Click **Create**, and then click **Close**.
6. Verify that **Action-SSL-OWA** appears in the **SSL Actions** page.

Creating SSL Policies

SSL policies are created by using the policy infrastructure. Each SSL policy has an SSL action bound to it, and the action is carried out when incoming traffic matches the rule that has been configured in the policy.

To create an SSL policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an SSL policy and verify the configuration:

- `add ssl policy <name> -rule <expression> -reqAction <string>`
- `show ssl policy <name>`

Example

```
> add ssl policy Policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA
Done
> show ssl policy Policy-SSL-1
Name: Policy-SSL-1      Rule: ns_true
Action: Action-SSL-OWA  Hits: 0
Policy is bound to following entities
1)      PRIORITY : 0
Done
```

To create an SSL policy by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create SSL Policy** dialog box, in the **Name** text box, type the name of the SSL Policy (for example, `Policy-SSL-1`).
4. In **Request Action**, select the configured SSL action that you want to associate with this policy (for example, **Action-SSL-OWA**). The `ns_true` general expression applies the policy to all successful SSL handshake traffic. However, if you need to filter specific responses, you can create policies with a higher level of detail. For more information about configuring granular policy expressions, see Understanding Policies and Expressions.
5. In **Named Expressions**, choose the built-in general expression `ns_true` and click **Add Expression**. The expression `ns_true` now appears in the Expression text box.
6. Click **Create**, and then click **Close**.
7. Verify that the policy is correctly configured by selecting the policy and viewing the **Details** section at the bottom of the pane.

Binding the SSL Policy to an SSL Vserver

After you configure an SSL policy for OWA, bind the policy to a vserver that will intercept incoming Outlook traffic. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

To bind an SSL policy to an SSL vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an SSL policy to an SSL vserver and verify the configuration:

- `bind ssl vserver <vServerName> -policyName <string>`
- `show ssl vserver <name>`

Example

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1
 Done
> show ssl vserver Vserver-SSL-1
        Advanced SSL configuration for VServer Vserver-SSL-1:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: ENABLED
        SSLv2 Redirect: ENABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1)      CertKey Name: CertKey-SSL-1 Server Certificate

1)      Policy Name: Policy-SSL-1
                        Priority: 0
1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
>
```

To bind an SSL policy to an SSL vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, select the vserver (for example, **Vserver-SSL-1**), and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, click **Insert Policy**, and then select the policy that you want to bind to the SSL vserver. Optionally, you can double-click the **Priority** field and type a new priority level.
4. Click **OK.**

Features at a Glance

Citrix® NetScaler® features can be configured independently or in combinations to address specific needs. Although some features fit more than one category, the numerous NetScaler features can generally be categorized as application switching and traffic management features, application acceleration features, application security and firewall features, and an application visibility feature.

To understand the order in which the features perform their processing, see Processing Order of Features.

Application Switching and Traffic Management Features

SSL Offloading

Transparently offloads SSL encryption and decryption from web servers, freeing server resources to service content requests. SSL places a heavy burden on an application's performance and can render many optimization measures ineffective. SSL offload and acceleration allow all the benefits of Citrix Request Switching technology to be applied to SSL traffic, ensuring secure delivery of web applications without degrading end-user performance.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Access Control Lists

Compares incoming packets to the Access Control Lists (ACLs). If a packet matches an ACL rule, the action specified in the rule is applied to the packet. Otherwise, the default action (ALLOW) is applied and the packet is processed normally by the system. For the system to compare incoming packets to the ACLs, you need to apply the ACLs. All ACLs are enabled by default, but you have to apply them in order for the NetScaler to compare incoming packets against them. If an ACL is not required to be a part of the lookup table, but still needs to be retained in the configuration, it should be disabled before the ACLs are applied. A NetScaler does not compare incoming packets to disabled ACLs.

For more information, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

Load Balancing

Load balancing decisions are based on a variety of algorithms, including round robin, least connections, weighted least bandwidth, weighted least packets, minimum response time, and hashing based on URL, domain source IP or destination IP. Both the TCP and UDP protocols are supported, so the NetScaler can load balance all traffic that uses those protocols as the underlying carrier (for example, HTTP, HTTPS, UDP, DNS, FTP, NNTP, and general firewall traffic). In addition, the NetScaler can maintain session persistence based on source IP, cookie, server, group or SSL session. It allows users to apply custom Extended Content Verification (ECV) to servers, caches, firewalls and other infrastructure devices to ensure that these systems are functioning properly and are providing the right content to users. It can also perform health checks using ping, TCP, or HTTP URL, and the user can create monitors based on Perl scripts.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Content Switching

Determines which server to send the request to based on the configured content switching policies. Policy rules can be configured based on the IP address, URL, and HTTP headers. This allows switching decisions to be based on user and device characteristics such as who the user is, what type of agent is being used, and what content the user requested.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Global Server Load Balancing (GSLB)

Extends the traffic management capabilities of a NetScaler to include distributed Internet sites and global enterprises. Whether installations are spread across multiple network locations or multiple clusters in a single location, the NetScaler maintains availability and distributes traffic across them. It makes intelligent DNS decisions to prevent users from being sent to a site that is down or overloaded. When the proximity-based GSLB method is enabled, the NetScaler can make load balancing decisions based on the proximity of the client's local DNS server (LDNS) in relation to different sites. The main benefit of the proximity-based GSLB method is faster response time resulting from the selection of the closest available site.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Dynamic Routing

Enables routers to obtain topology information, routes, and IP addresses from neighboring routers automatically. When dynamic routing is enabled, the corresponding routing process listens to route updates and advertises routes. The routing processes can also be placed in passive mode. Routing protocols enable an upstream router to load balance traffic to identical vservers hosted on two standalone NetScaler units using the Equal Cost Multipath technique.

For more information, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

Link Load Balancing

Load balances multiple WAN links and provides link failover, further optimizing network performance and ensuring business continuity. Ensures that network connections remain highly available, by applying intelligent traffic control and health checks to distribute traffic efficiently across upstream routers. Identifies the best WAN link to route both incoming and outbound traffic based on policies and network conditions, and protects applications against WAN or Internet link failure by providing rapid fault detection and failover.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

TCP Optimization

Transfers certain TCP tasks from your managed servers to the NetScaler, reducing CPU load on your managed servers and improving performance.

For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Web Interface on NetScaler

Provides access to XenApp and XenDesktop resources, which include applications, content, and desktops. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in. The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance.

Note: Web Interface is supported only on NetScaler nCore releases.

For more information, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Load Balancing of Branch Repeaters for WAN Optimization

To provide high-scale WAN optimization, the branch repeaters deployed at data centers can be load balanced through NetScaler appliances. The bandwidth and number of concurrent sessions can be improved significantly.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

OpenCloud Bridge

The Citrix NetScaler® OpenCloud Bridge™ feature, a fundamental part of the Citrix® OpenCloud framework, is a tool used to build a cloud-extended data center. The OpenCloud Bridge enables you to connect one or more NetScaler appliances or VPXs on the cloud-to your network without reconfiguring your network. Cloud hosted applications appear as though they are running on one contiguous enterprise network.The primary purpose of the OpenCloud Bridge is to enable companies to move their applications to the cloud while reducing costs and the risk of application failure. In addition, the OpenCloud Bridge increases network security in cloud environments. An OpenCloud Bridge is a Layer-2 network bridge that connects a NetScaler appliance or VPX on a cloud instance to a NetScaler appliance or VPX on your LAN. The connection is made through a tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. Then Internet Protocol security (IPsec) protocol suite is used to secure the communication between the peers in the OpenCloud Bridge.

For more information, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

DataStream™

The DataStream feature of NetScaler provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a

decision based on SQL query parameters, such as user name and database name, command parameters. You can further configure monitors to track the state of database servers.

The advanced policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with `MYSQL.CLIENT` and `MYSQL.REQ`) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with `MYSQL.RES`) to evaluate server responses to user-configured health monitors. For more information about these expressions, see the "Evaluating Connections to Database Servers" chapter in the *Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: DataStream is supported only for MySQL databases. For information about the DataStream feature, see the "DataStream" chapter in the *Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Application Acceleration Features

AppCompress

Provides transparent compression for HTML and text files using the gzip compression protocol. The typical 4:1 compression ratio yields up to 50% reduction in bandwidth requirements out of the data center. This also results in significantly improved end-user response time by reducing the amount of data that must be delivered to the user's browser.

For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Cache Redirection

Manages the flow of traffic to a reverse proxy, transparent proxy, or forward proxy cache farm. Inspects all requests, and identifies non-cacheable requests and sends them directly to the origin servers over persistent connections. By intelligently redirecting non-cacheable requests back to the origin web servers, the NetScaler frees cache resources and increases cache hit rates while reducing overall bandwidth consumption and response delays for these requests.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

AppCache

Helps optimize web content and application data delivery by providing a fast in-memory HTTP/1.1 and HTTP/1.0 compliant web caching for both static and dynamic content. This on-board cache stores the results of incoming application requests even when an incoming request is secured or the data compressed, and then reuses the data to fulfill subsequent requests for the same information. By serving data directly from the on-board cache, the NetScaler can reduce page regeneration times by eliminating the need to funnel static and dynamic content requests to the server.

For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

TCP Buffering

Buffers the server's response and delivers it to the client at the client's speed, thus offloading the server faster and thereby improving the performance of web sites. For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Application Security and Firewall Features

Denial of Service Attack (DoS) Defense

Detects and stops malicious distributed denial-of-service (DDoS) attacks and other types of malicious attacks before they reach your servers, preventing them from affecting network and application performance. The NetScaler identifies legitimate clients and elevates their priority, leaving suspect clients unable to consume a disproportionate percentage of resources and cripple your site. A NetScaler provides application-level protection from the following types of malicious attacks:

- SYN flood attacks
- Pipeline attacks
- Teardop attacks
- Land attacks
- Fraggle attacks
- Zombie connection attacks

The NetScaler aggressively defends against these types of attacks by preventing the allocation of server resources for these connections. This insulates servers from the overwhelming flood of packets associated with these events.

The NetScaler also protects network resources from ICMP based attacks by using ICMP rate limiting and aggressive ICMP packet inspection. It performs strong IP reassembly, drops a variety of suspicious and malformed packets, and applies Access Control Lists (ACLs) to site traffic for further protection.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Content Filtering

Provides protection from malicious attacks for web sites at the Layer 7 level. The NetScaler inspects each incoming request according to user-configured rules based on HTTP headers, and performs the action the user configured. Actions can include resetting the connection, dropping the request, or sending an error message to the user's browser. This allows the NetScaler to screen unwanted requests and reduces your servers' exposure to attacks.

This feature can also analyze HTTP GET and POST requests and filter out known bad signatures, allowing it to defend your servers against HTTP-based attacks such as variants of the Nimda and Code Red viruses.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Responder

Functions like an advanced filter and can be used to generate responses from the NetScaler to the client. Some common uses of this feature are generation of redirect responses, user defined responses or resets.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Rewrite

Modifies HTTP headers and body text. You can use it to add HTTP headers to an HTTP request or response, make modifications to individual HTTP headers, or delete HTTP headers. It also lets you modify the HTTP body in requests and responses.

When the NetScaler receives a request or sends a response, it checks for rewrite rules, and if applicable rules exist, it applies them to the request or response before passing it on to the web server or client computer.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Priority Queuing

Prioritizes user requests to ensure that the most important traffic is serviced first during surges in request volume. You can establish priority based on request URLs, cookies, or a variety of other factors. The NetScaler places requests in a three-tier queue based on their configured priority, enabling business-critical transactions to flow smoothly even during surges or site attacks.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Surge Protection

Regulates the flow of user requests to servers and controls the number of users that can simultaneously access the resources on the servers, queuing any additional requests once your servers have reached their capacity. By controlling the rate at which connections can be established, the NetScaler blocks surges in requests from being passed on to your servers, thus preventing site overload.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Access Gateway

Citrix Access Gateway is a secure application access solution that provides administrators granular application-level policy and action controls to secure access to applications and data while allowing users to work from anywhere. It gives IT administrators a single point of control and tools to help ensure compliance with regulations and the highest levels of information security across and outside the enterprise. At the same time, it empowers users with a single point of access—optimized for roles, devices, and networks—to the

enterprise applications and data they need. This unique combination of capabilities helps maximize the productivity of today's mobile workforce.

For more information, see Access Gateway.

Application Firewall

Protects applications from misuse by hackers and malware, such as cross site scripting attacks, buffer overflow attacks, SQL injection attacks, and forceful browsing, by filtering traffic between each protected web server and users that connect to any web site on that web server. The Application Firewall examines all traffic for evidence of attacks on web server security or misuse of web server resources, and takes the appropriate action to prevent these attacks from succeeding.

For more information, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Application Visibility Feature

EdgeSight for NetScaler

Support for application performance monitoring based on end user experience. This solution leverages the HTML injection feature to obtain various time values, which are used by EdgeSight server for analysis and report generation. EdgeSight for NetScaler provides a way to monitor the performance benefits of a NetScaler and determine potential bottlenecks in a network.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Enhanced Application Visiblity Using AppFlow

The Citrix® NetScaler® appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

Note: The AppFlow feature is supported only by Citrix® nCore™ software.

Getting Started with Citrix NetScaler VPX

Intended for system and network administrators who install and configure complex networking equipment, this section of the library describes initial setup and basic configuration of the Citrix® NetScaler® VPX™ product, including the following topics.

Citrix NetScaler VPX Overview	Defines NetScaler VPX and includes a description of the virtualization platforms on which NetScaler VPX can be hosted.
Understanding the NetScaler	What a NetScaler is and where it fits in a network, with descriptions of entities used in typical configurations and the order in which data is processed by the various features.
Installing NetScaler Virtual Appliances on XenServer	Prerequisites and tasks for installing NetScaler VPX on XenServer.
Installing NetScaler Virtual Appliances on VMware ESX	Prerequisites and tasks for installing NetScaler VPX on VMware ESX 4.0 and VMware ESX 3.5.
Configuring the Basic System Settings	Tasks for setting up an initial configuration by using the NetScaler VPX Console in XenCenter and tasks for configuring a NetScaler VPX virtual appliance.
Understanding Common Network Topologies	Describes the four common deployment topologies: Two-Arm Multiple Subnet, Two-Arm Transparent, One-Arm Single Subnet, and One-Arm Multiple Subnet. Includes topology diagrams, sample values, and references.
Configuring System Management Settings	Procedures for configuring basic system management settings, such as VLANs, SNMP, and DNS.
Load Balancing Traffic on a Citrix NetScaler	Basic introduction to the load balancing feature. Includes procedures for configuring a basic load balancing setup to deliver a Web application, and procedures for configuring persistence, URL redirection, and backup vservers.
Accelerating Load Balanced Traffic by Using Compression	Basic introduction to the compression feature. Includes procedures for configuring a NetScaler to compress application traffic.

Securing Load Balanced Traffic by Using SSL	Basic introduction to the SSL offload feature. Includes procedures for configuring a NetScaler to secure application traffic by using SSL.
Features at a Glance	Brief description of all the features, with links to documentation for the features.

Citrix NetScaler VPX Overview

The Citrix® NetScaler® VPX™ product is a virtual NetScaler appliance that can be hosted on Citrix XenServer® and VMware ESX or ESXi, and Microsoft Hyper-V virtualization platforms.

A NetScaler virtual appliance installed on the Citrix XenServer or Microsoft Server 2008 R2 supports all the features of a physical NetScaler, except interface-related events and tagged VLANs. A NetScaler virtual appliance installed on the VMware ESX platform does not support interface-related events, but does support tagged VLANs. For the VLAN tagging feature to work, you must set the port group's VLAN ID to 4095 on the VSwitch of VMware ESX server. For more information about setting a VLAN ID on the VSwitch of VMware ESX server, see http://www.vmware.com/.

This overview covers only aspects that are unique to NetScaler VPX. For an overview of NetScaler VPX functionality, see Understanding the NetScaler.

Note: The terms *NetScaler*, *NetScaler appliance*, and *appliance* are used interchangeably with *NetScaler virtual appliance* unless stated otherwise.

Citrix NetScaler VPX Overview

The Citrix® NetScaler® VPX™ product is a virtual NetScaler appliance that can be hosted on Citrix XenServer® and VMware ESX or ESXi, and Microsoft Hyper-V virtualization platforms.

A NetScaler virtual appliance installed on the Citrix XenServer or Microsoft Server 2008 R2 supports all the features of a physical NetScaler, except interface-related events and tagged VLANs. A NetScaler virtual appliance installed on the VMware ESX platform does not support interface-related events, but does support tagged VLANs. For the VLAN tagging feature to work, you must set the port group's VLAN ID to 4095 on the VSwitch of VMware ESX server. For more information about setting a VLAN ID on the VSwitch of VMware ESX server, see http://www.vmware.com/.

This overview covers only aspects that are unique to NetScaler VPX. For an overview of NetScaler VPX functionality, see Understanding the NetScaler.

Note: The terms *NetScaler*, *NetScaler appliance*, and *appliance* are used interchangeably with *NetScaler virtual appliance* unless stated otherwise.

NetScaler VPX Setup for the XenServer Platform

When you set up NetScaler® VPX™ on XenServer®, you must use the XenCenter® client to install the first NetScaler virtual appliance. Subsequent virtual appliances can be added by using either the XenCenter client or Citrix® Command Center.

XenServer

The XenServer® product is a server virtualization platform that offers near bare-metal virtualization performance for virtualized server and client operating systems. XenServer uses the Xen® hypervisor to virtualize each server on which it is installed, enabling each server to host multiple virtual machines simultaneously.

The following figure shows the bare-metal solution architecture of NetScaler® VPX™ on XenServer.

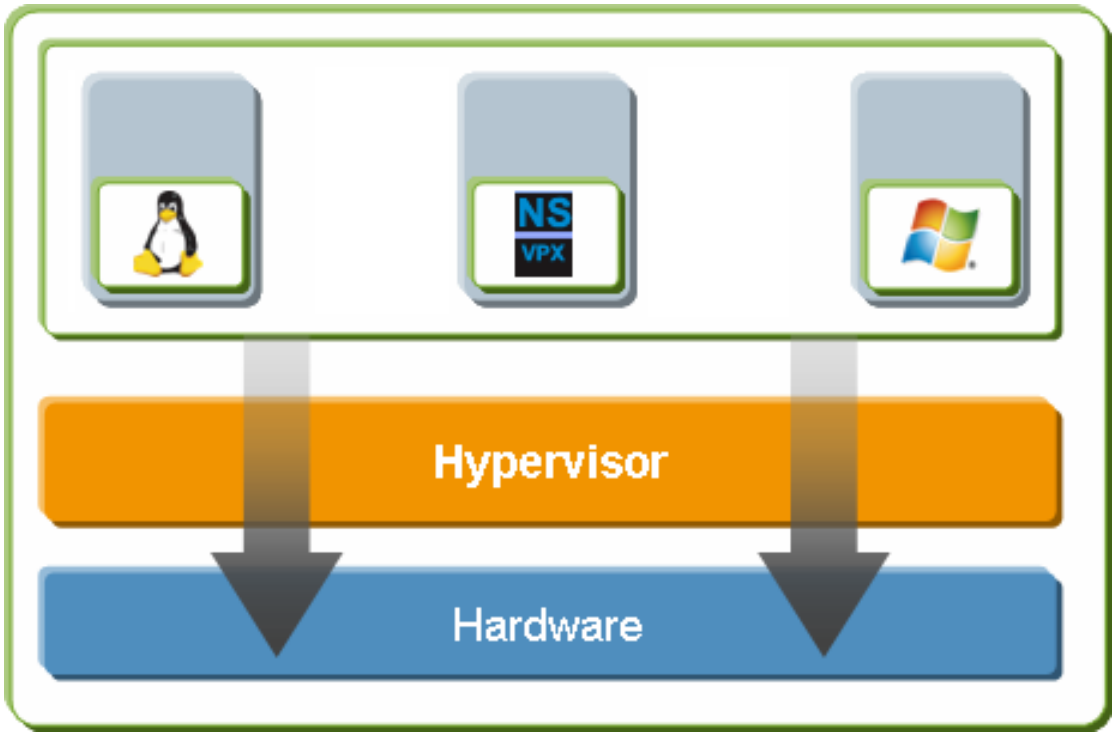


Figure 1. NetScaler VPX on XenServer

The bare-metal solution architecture has the following components:

Hardware or physical layer:

Physical hardware components including memory, CPU, network cards, and disk drives.

Xen hypervisor:

Thin layer of software that runs on top of the hardware. The Xen hypervisor gives each virtual machine a dedicated view of the hardware.

Virtual machine:

Operating system hosted on the hypervisor and appearing to the user as a separate physical computer. However, the machine shares physical resources with other virtual machines, and it is portable because the virtual machine is abstracted from physical hardware.

A NetScaler VPX virtual machine, or *virtual appliance*, is installed on the Xen hypervisor and uses paravirtualized drivers to access storage and network resources. It appears to the users as an independent NetScaler appliance with its own network identity, user authorization and authentication capabilities, configuration, applications, and data. The paravirtualization technique enables the virtual machines and the hypervisor to work together to achieve high performance for I/O and for CPU and memory virtualization.

For more information about XenServer, see the XenServer documentation at http://support.citrix.com/product/xens/.

XenCenter

XenCenter® is a graphical virtualization-management interface for XenServer®, enabling you to manage servers, resource pools, and shared storage, and to deploy, manage, and monitor virtual machines from your Windows desktop machine.

Use XenCenter to install NetScaler VPX on XenServer.

For more information about XenCenter, see the XenServer documentation at http://support.citrix.com/product/xens/.

Command Center

Command Center is a management and monitoring solution for Citrix application networking products that include NetScaler, NetScaler VPX, Citrix Access Gateway™ Enterprise Edition, Citrix® Branch Repeater™, Branch Repeater VPX™, and Citrix Repeater™. Command Center enables network administrators and operations teams to manage, monitor, and troubleshoot the entire global application delivery infrastructure from a single, unified console.

This centralized management solution simplifies operations by providing administrators with enterprise-wide visibility and automating management tasks that need to be executed across multiple devices.

Command Center is available with Citrix NetScaler Enterprise and Platinum editions.

You can use Command Center to provision NetScaler VPX on XenServer, and then you can manage and monitor the virtual appliances from Command Center.

Note: You must use the XenCenter client to manage XenServer. You cannot manage XenServer from Command Center.

For more information about Command Center, see the Command Center documentation.

NetScaler VPX Setup for the VMware ESX Platform

The NetScaler® VPX™ setup for the VMware ESX platform requires a VMware ESX or ESXi server and the vSphere client.

VMware ESX and ESXi are virtualization products based on bare-metal architecture, offered by VMware, Inc. Citrix NetScaler VPX can be hosted on a VMware ESX or ESXi server.

For more information about VMware ESX, see http://www.vmware.com/.

The vSphere client is a graphical interface for managing virtual machines on VMware ESX servers. You use the vSphere client to allocate resources on the ESX server to virtual appliances installed on the server or to deallocate resources. For example, you can allocate virtual network ports to a virtual appliance.

For more information about VMware vSphere client, see http://www.vmware.com/.

Understanding the NetScaler

The Citrix® NetScaler® product is an application switch that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4-L7) network traffic for web applications. For example, a NetScaler makes load balancing decisions on individual HTTP requests rather than on the basis of long-lived TCP connections, so that the failure or slowdown of a server is managed much more quickly and with less disruption to clients. The NetScaler feature set can be broadly categorized as consisting of switching features, security and protection features, and server-farm optimization features.

Switching Features

When deployed in front of application servers, a NetScaler ensures optimal distribution of traffic by the way in which it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP or TCP request, and on the basis of L4-L7 header information such as URL, application data type, or cookie. Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

Security and Protection Features

NetScaler security and protection features protect web applications from application-layer attacks. A NetScaler allows legitimate client requests and can block malicious requests. It provides built-in defenses against denial-of-service (DoS) attacks and supports features that protect applications against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from application-layer attacks, including buffer overflow exploits, SQL injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

Optimization Features

Optimization features offload resource-intensive operations such as Secure Sockets Layer (SSL) processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. This improves the performance of the servers in the server farm and therefore speeds up applications. A NetScaler supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

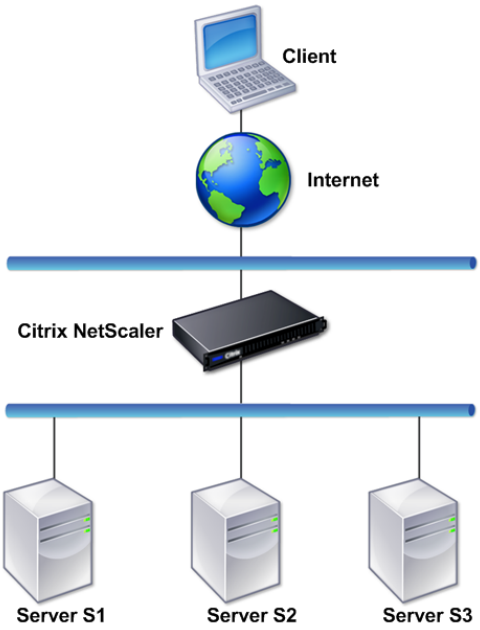
Where Does a NetScaler Fit in the Network?

A NetScaler resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers (vservers) configured on the NetScaler provide connection points that clients use to access the applications behind the NetScaler. In this case, the NetScaler owns public IP addresses that are associated with its vservers, while the real servers are isolated in a private network. It is also possible to operate the NetScaler in a transparent mode as an L2 bridge or L3 router, or even to combine aspects of these and other modes.

Physical Deployment Modes

A NetScaler logically residing between clients and servers can be deployed in either of two physical modes: inline and one-arm. In inline mode, multiple network interfaces are connected to different Ethernet segments, and the NetScaler is placed between the clients and the servers. The NetScaler has a separate network interface to each client network and a separate network interface to each server network. The NetScaler and the servers can exist on different subnets in this configuration. It is possible for the servers to be in a public network and the clients to directly access the servers through the NetScaler, with the NetScaler transparently applying the L4-L7 features. Usually, vservers (described later) are configured to provide an abstraction of the real servers. The following figure shows a typical inline deployment.

Figure 1. Inline Deployment



In one-arm mode, only one network interface of the NetScaler is connected to an Ethernet segment. The NetScaler in this case does not isolate the client and server sides of the network, but provides access to applications through configured vservers. One-arm mode can simplify network changes needed for NetScaler installation in some environments.

For examples of inline (two-arm) and one-arm deployment, see Understanding Common Network Topologies.

Citrix NetScaler as an L2 Device

A NetScaler functioning as an L2 device is said to operate in L2 mode. In L2 mode, the NetScaler forwards packets between network interfaces when all of the following conditions are met:

- The packets are destined to another device's media access control (MAC) address.
- The destination MAC address is on a different network interface.
- The network interface is a member of the same virtual LAN (VLAN).

By default, all network interfaces are members of a pre-defined VLAN, VLAN 1. Address Resolution Protocol (ARP) requests and responses are forwarded to all network interfaces that are members of the same VLAN. To avoid bridging loops, L2 mode must be disabled if another L2 device is working in parallel with the NetScaler.

For information about how the L2 and L3 modes interact, see Configuring Modes of Packet Forwarding.

For information about configuring L2 mode, see Enabling and Disabling Layer 2 Mode.

Citrix NetScaler as a Packet Forwarding Device

A NetScaler can function as a packet forwarding device, and this mode of operation is called L3 mode. With L3 mode enabled, the NetScaler forwards any received unicast packets that are destined for an IP address that it does not have internally configured, if there is a route to the destination. A NetScaler can also route packets between VLANs.

In both modes of operation, L2 and L3, a NetScaler generally drops packets that are in:

- Multicast frames
- Unknown protocol frames destined for a NetScaler's MAC address (non-IP and non-ARP)
- Spanning Tree protocol (unless BridgeBPDUs is ON)

For information about how the L2 and L3 modes interact, see Configuring Modes of Packet Forwarding.

For information about configuring the L3 mode, see Enabling and Disabling Layer 3 Mode.

How a NetScaler Communicates with Clients and Servers

A NetScaler is usually deployed in front of a server farm and functions as a transparent TCP proxy between clients and servers, without requiring any client-side configuration. This basic mode of operation is called Request Switching technology and is the core of NetScaler functionality. Request Switching enables a NetScaler to multiplex and offload the TCP connections, maintain persistent connections, and manage traffic at the request (application layer) level. This is possible because the NetScaler can separate the HTTP request from the TCP connection on which the request is delivered.

Depending on the configuration, a NetScaler may process the traffic before forwarding the request to a server. For example, if the client attempts to access a secure application on the server, the NetScaler might perform the necessary SSL processing before sending traffic to the server.

To facilitate efficient and secure access to server resources, a NetScaler uses a set of IP addresses collectively known as NetScaler-owned IP addresses. To manage your network traffic, you assign NetScaler-owned IP addresses to virtual entities that become the building blocks of your configuration. For example, to configure load balancing, you create virtual servers (vservers) to receive client requests and distribute them to services, which are entities representing the applications on your servers.

Understanding NetScaler-Owned IP Addresses

To function as a proxy, a NetScaler uses a variety of IP addresses. The key NetScaler-owned IP addresses are:

NetScaler IP address (NSIP)

The NSIP is the IP address for management and general system access to the NetScaler itself, and for HA communication.

Mapped IP address (MIP)

A MIP is used for server-side connections. It is not the IP address of the NetScaler. In most cases, when the NetScaler receives a packet, it replaces the source IP address with a MIP before sending the packet to the server. With the servers abstracted from the clients, the NetScaler manages connections more efficiently.

Virtual server IP address (VIP)

A VIP is the IP address associated with a vserver. It is the public IP address to which clients connect. A NetScaler managing a wide range of traffic may have many VIPs configured.

Subnet IP address (SNIP)

When the NetScaler is attached to multiple subnets, SNIPs can be configured for use as MIPs providing access to those subnets. SNIPs may be bound to specific VLANs and interfaces.

IP Set

An IP set is a set of IP addresses, which are configured on the NetScaler appliance as SNIPs or MIPs. An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it.

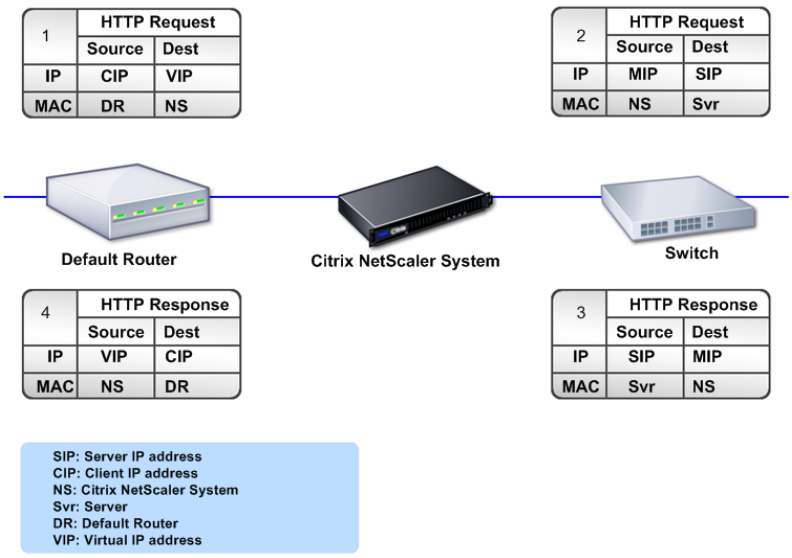
Net Profile

A net profile (or network profile) contains an IP address or an IP set. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. During communication with physical servers or peers, the NetScaler appliance uses the addresses specified in the profile as the source IP address.

How Traffic Flows Are Managed

Because a NetScaler functions as a TCP proxy, it translates IP addresses before sending packets to a server. When you configure a vserver, clients connect to a VIP on the NetScaler instead of directly connecting to a server. Based on the settings on the vserver, the NetScaler selects an appropriate server and sends the client's request to that server. By default, the NetScaler uses a MIP or SNIP to establish connections with the server, as shown in the following figure.

Figure 1. Vserver-Based Connections



Note: You can use SNIP instead of MIP in the preceding figure.

In the absence of a vserver, when a NetScaler receives a request, it transparently forwards the request to the server. This is called the transparent mode of operation. When operating in transparent mode, a NetScaler translates the source IP addresses of incoming client requests to the MIP or SNIP but does not change the destination IP address. For this mode to work, L2 or L3 mode needs to be configured appropriately.

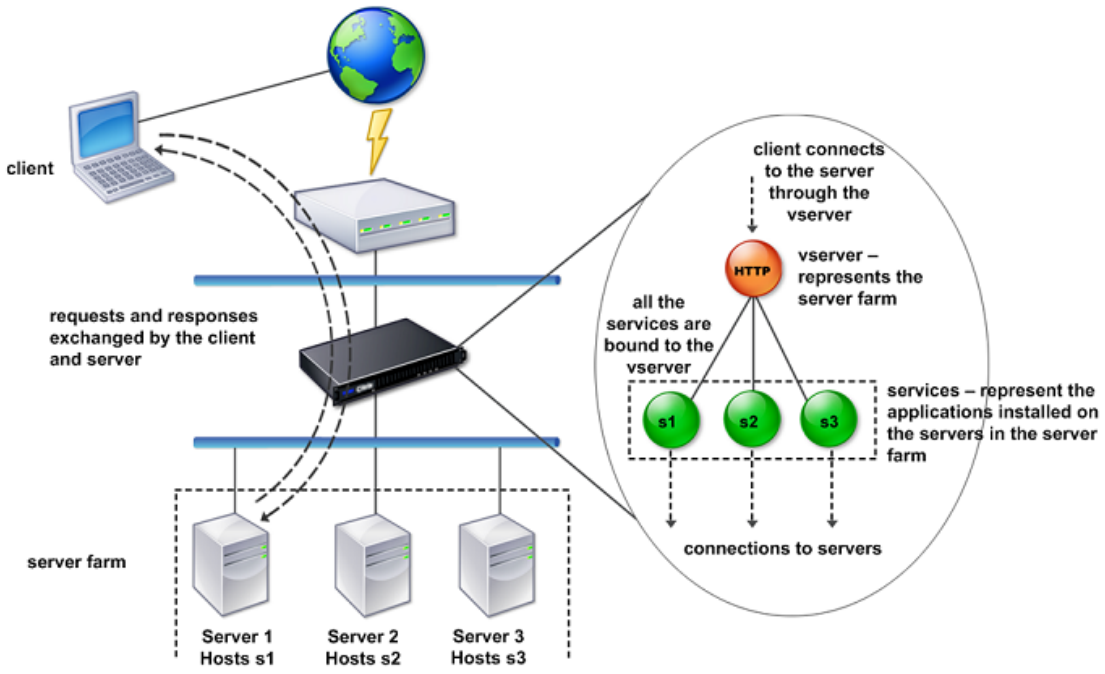
For cases in which the servers need the actual client IP address, the NetScaler can be configured to modify the HTTP header by inserting the client IP address as an additional field, or configured to use the client IP address instead of the MIP or SNIP for connections to the servers.

Traffic Management Building Blocks

The configuration of a NetScaler is typically built up with a series of virtual entities that serve as building blocks for traffic management. The building block approach helps separate traffic flows. Virtual entities are abstractions, typically representing IP addresses, ports, and protocol handlers for processing traffic. Clients access applications and resources through these virtual entities. The most commonly used entities are vservers and services. Vservers represent groups of servers in a server farm or remote network, and services represent specific applications on each server.

Most features and traffic settings are enabled through virtual entities. For example, you can configure a NetScaler to compress all server responses to a client that is connected to the server farm through a particular vserver. To configure the NetScaler for a particular environment, you need to identify the appropriate features and then choose the right mix of virtual entities to deliver them. Most features are delivered through a cascade of virtual entities that are bound to each other. In this case, the virtual entities are like blocks being assembled into the final structure of a delivered application. You can add, remove, modify, bind, enable, and disable the virtual entities to configure the features. The following figure shows the concepts covered in this section.

Figure 1. How Traffic Management Building Blocks Work

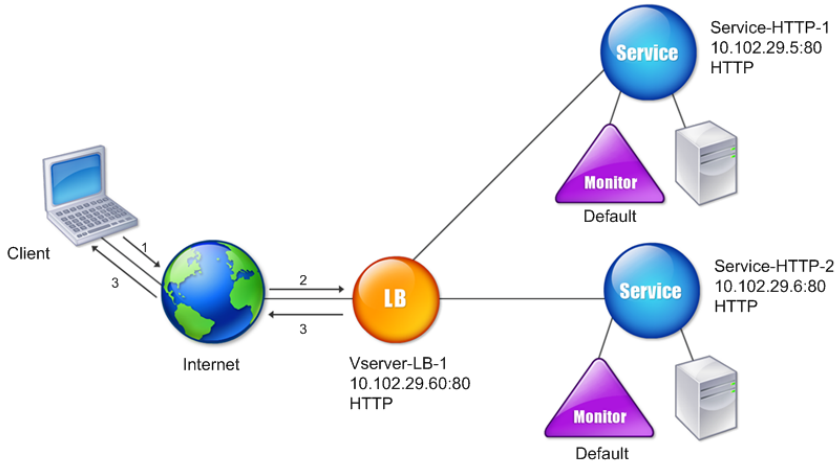


A Simple Load Balancing Configuration

In the example shown in the following figure, the NetScaler is configured to function as a load balancer. For this configuration, you need to configure virtual entities specific to load balancing and bind them in a specific order. As a load balancer, a NetScaler distributes client requests across several servers and thus optimizes the utilization of resources.

The basic building blocks of a typical load balancing configuration are services and load balancing vservers. The services represent the applications on the servers. The vservers abstract the servers by providing a single IP address to which the clients connect. To ensure that client requests are sent to a server, you need to bind each service to a vserver. That is, you must create services for every server and bind the services to a vserver. Clients use the VIP to connect to a NetScaler. When the NetScaler receives client requests on the VIP, it sends them to a server determined by the load balancing algorithm. Load balancing uses a virtual entity called a monitor to track whether a specific configured service (server plus application) is available to receive requests.

Figure 1. Load Balancing Virtual Server, Services, and Monitors



In addition to configuring the load balancing algorithm, you can configure several parameters that affect the behavior and performance of the load balancing configuration. For example, you can configure the vserver to maintain persistence based on source IP address. The NetScaler then directs all requests from any specific IP address to the same server.

Understanding Virtual Servers

A vserver is a named NetScaler entity that external clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP address (VIP), port, and protocol. The name of the vserver is only of local significance and is designed to make the vserver easier to identify. When a client attempts to access applications on a server, it sends a request to the VIP instead of the IP address of the physical server. When the NetScaler receives a request on the VIP, it terminates the connection at the vserver and uses its own connection with the server on behalf of the client. The port and protocol settings of the vserver determine the applications that the vserver represents. For example, a web server can be represented by a vserver and a service whose port and protocol are set to 80 and HTTP, respectively. Multiple vservers can use the same VIP but different protocols and ports.

Vservers are points for delivering features. Most features, like compression, caching, and SSL offload, are normally enabled on a vserver. When the NetScaler receives a request on a VIP, it chooses the appropriate vserver by the port on which the request was received and its protocol. The NetScaler then processes the request as appropriate for the features configured on the vserver.

In most cases, vservers work in tandem with services. You can bind multiple services to a vserver. These services represent the applications running on physical servers in a server farm. After the NetScaler processes requests received on a VIP, it forwards them to the servers as determined by the load balancing algorithm configured on the vserver. The following figure shows these concepts.

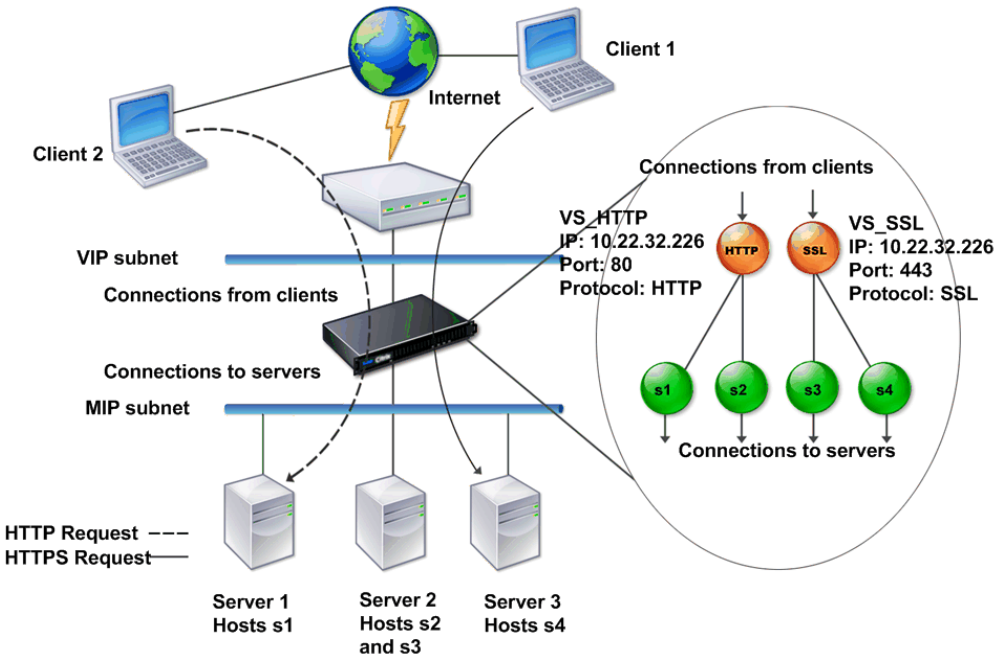


Figure 1. Multiple Virtual Servers on a Single VIP

The preceding figure shows a configuration consisting of two vservers with a common VIP but different ports and protocols. Each of these vservers has two services bound to it. The services s1 and s2 are bound to VS_HTTP and represent the HTTP applications on Server 1 and Server 2. The services s3 and s4 are bound to VS_SSL and represent the SSL applications on Server 2 and Server 3 (Server 2 provides both HTTP and SSL applications). When the NetScaler receives an HTTP request on the VIP, it processes the request based on the settings of VS_HTTP and sends it to either Server 1 or Server 2. Similarly, when the NetScaler receives an HTTPS request on the VIP, it processes it based on the settings of VS_SSL and it sends it to either Server 2 or Server 3.

Vservers are not always represented by specific IP address, port numbers, or protocols. They can be represented by wildcards, in which case they are known as wildcard vservers. For example, when you configure a vserver with a wildcard instead of a VIP, but with a specific port number, the NetScaler intercepts and processes all traffic conforming to that protocol and destined for the predefined port. For vservers with wildcards instead of VIPs and port numbers, the NetScaler intercepts and processes all traffic conforming to the protocol.

Vservers can be grouped into the following categories:

Load balancing virtual server

Receives and redirects requests to an appropriate server. Choice of the appropriate server is based on which of the various load balancing methods the user configures.

Cache redirection virtual server

Redirects client requests for dynamic content to origin servers and static content to cache servers. Cache redirection vservers often work in conjunction with load balancing vservers.

Content switching virtual server

Directs traffic to a server on the basis of the content that the client has requested. For example, you can create a content switching vserver that directs all client requests for images to a server that serves images only. Content switching vservers often work in conjunction with load balancing vservers.

Virtual private network (VPN) virtual server

Decrypts tunneled traffic and sends it to intranet applications.

SSL virtual server

Receives and decrypts SSL traffic, and then redirects to an appropriate server. Choosing the appropriate server is similar to choosing a load balancing virtual server.

Understanding Services

Services represent applications on a server. While services are normally combined with vservers, in the absence of a vserver, a service can still manage application-specific traffic. For example, you can create an HTTP service on a NetScaler to represent a web server application. When the client attempts to access a web site hosted on the web server, the NetScaler intercepts the HTTP requests and creates a transparent connection with the web server.

In service-only mode, a NetScaler functions as a proxy. It terminates client connections, uses a SNIP or MIP to establish a connection to the server, and translates incoming client requests to the SNIP or MIP. Although the clients send requests directly to the IP address of the server, the server sees them as coming from the SNIP or MIP. The NetScaler translates the IP addresses, port numbers, and sequence numbers.

A service is also a point for applying features. Consider the example of SSL acceleration. To use this feature, you must create an SSL service and bind an SSL certificate to the service. When the NetScaler receives an HTTPS request, it decrypts the traffic and sends it, in clear text, to the server. Only a limited set of features can be configured in the service-only case.

Services use entities called monitors to track the health of applications. Every service has a default monitor, which is based on the service type, bound to it. As specified by the settings configured on the monitor, the NetScaler sends probes to the application at regular intervals to determine its state. If the probes fail, the NetScaler marks the service as down. In such cases, the NetScaler responds to client requests with an appropriate error message or re-routes the request as determined by the configured load balancing policies.

Understanding Policies and Expressions

A policy defines specific details of traffic filtering and management on a NetScaler. It consists of two parts: the expression and the action. The expression defines the types of requests that the policy matches. The action tells the NetScaler what to do when a request matches the expression. As an example, the expression might be to match a specific URL pattern to a type of security attack, with the action being to drop or reset the connection. Each policy has a priority, and the priorities determine the order in which the policies are evaluated.

When a NetScaler receives traffic, the appropriate policy list determines how to process the traffic. Each policy on the list contains one or more expressions, which together define the criteria that a connection must meet to match the policy.

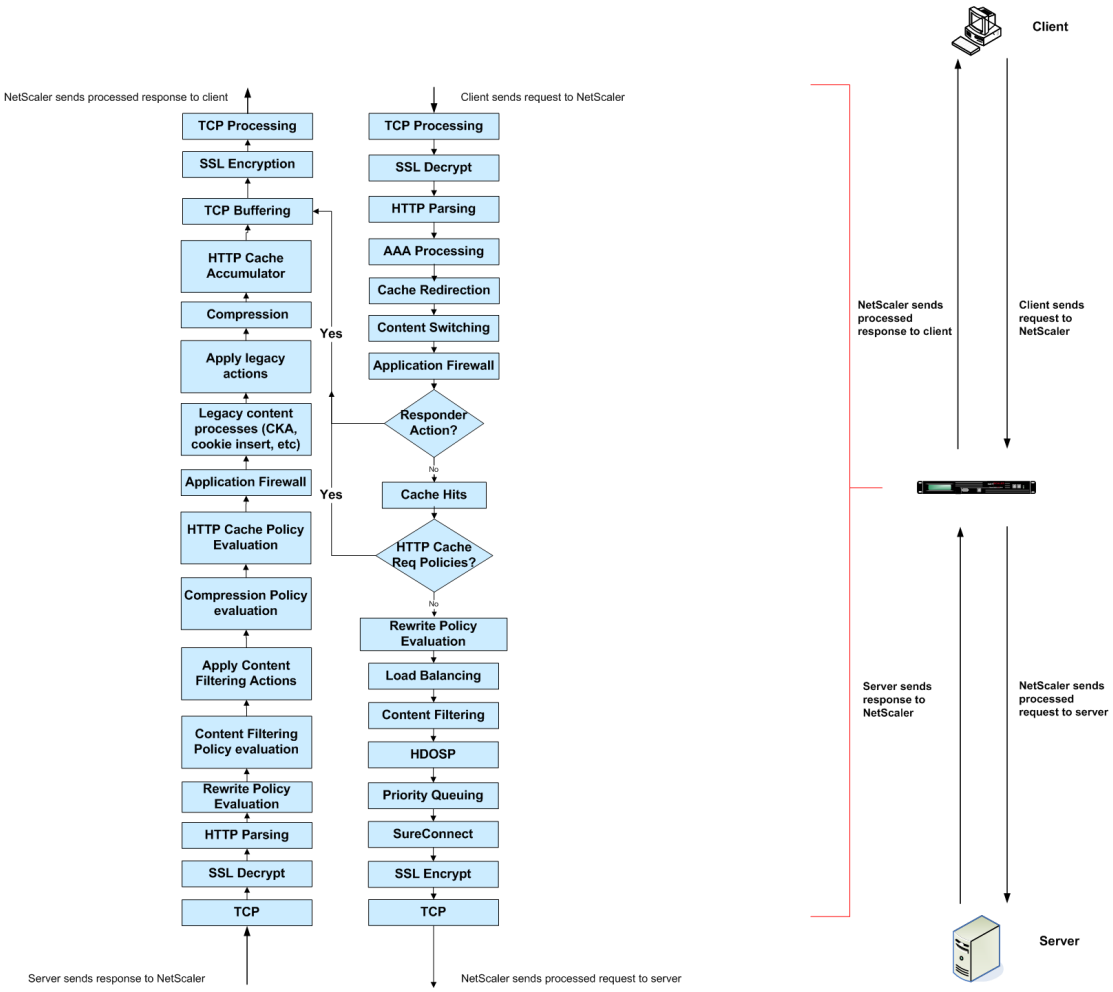
For all policy types except Rewrite policies, a NetScaler implements only the first policy that a request matches, not any additional policies that it might also match. For Rewrite policies, the NetScaler evaluates the policies in order and, in the case of multiple matches, performs the associated actions in that order. Policy priority is important for getting the results you want.

Processing Order of Features

Depending on requirements, you can choose to configure multiple features. For example, you might choose to configure both compression and SSL offload. As a result, an outgoing packet might be compressed and then encrypted before being sent to the client.

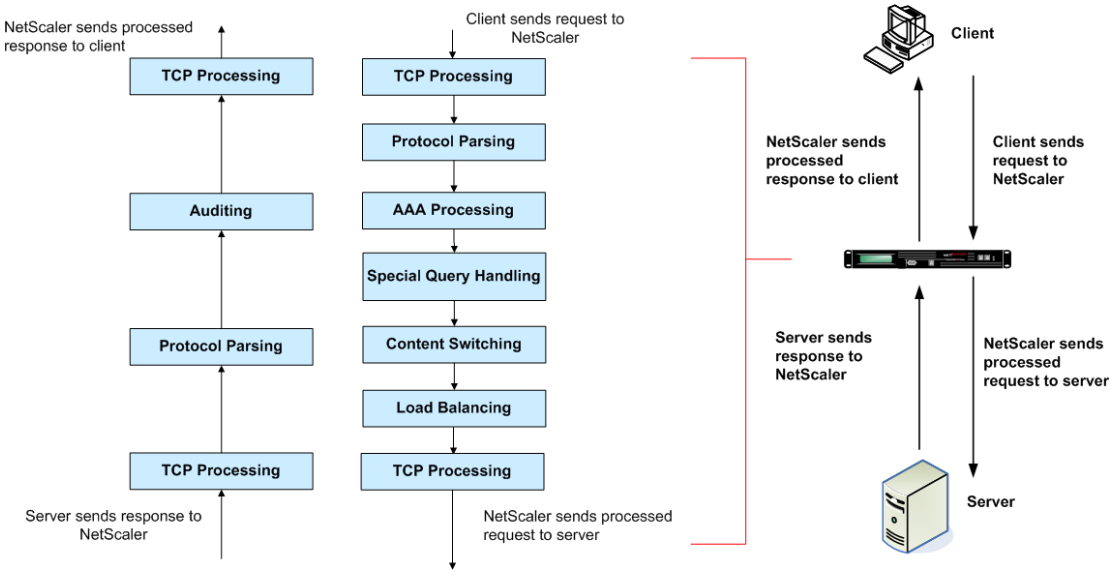
The following figure shows the L7 packet flow in the NetScaler.

Figure 1. L7 Packet Flow Diagram



The following figure shows the DataStream packet flow in the NetScaler. DataStream is supported only for MySQL databases. For information about the DataStream feature, see DataStream.

Figure 2. DataStream Packet Flow Diagram


NetScaler sends processed response to client
TCP Processing
Auditing
Protocol Parsing
TCP Processing
Server sends response to NetScaler
Client sends request to NetScaler
TCP Processing
Protocol Parsing
AAA Processing
Special Query Handling
Content Switching
Load Balancing
TCP Processing
NetScaler sends processed request to server
Client
NetScaler sends processed response to client
Client sends request to NetScaler
Server sends response to NetScaler
NetScaler sends processed request to server
Server

Installing NetScaler Virtual Appliances on XenServer

To install Citrix® NetScaler® virtual appliances on Citrix® XenServer®, you must first install XenServer on a machine with adequate system resources. To perform the NetScaler VPX installation, you use Citrix® XenCenter®, which must be installed on a remote machine that can connect to the XenServer host through the network.

Note: After the initial configuration of the NetScaler appliance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Prerequisites for Installing NetScaler Virtual Appliances on XenServer

Before you begin installing a virtual appliance, do the following:

- Install XenServer® version 5.6 or later on hardware that meets the minimum requirements.
- Install XenCenter® on a management workstation that meets the minimum system requirements.

  Obtain VPX license files. For more information about VPX licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx122426.

XenServer Hardware Requirements

The following table describes the minimum hardware requirements for a XenServer platform running NetScaler nCore VPX.

Table 1. Minimum System Requirements for XenServer Running NetScaler nCore VPX

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT or AMD-V) enabled **Note:** To run NetScaler VPX, hardware support for virtualization must be enabled on the XenServer host. Make sure that the BIOS option for virtualization support is not disabled. Consult your BIOS documentation for more details.
RAM	3 gigabytes (GB)
Disk space	Locally attached storage (PATA, SATA, SCSI) with 40 GB of disk space **Note:** XenServer installation creates a 4 GB partition for the XenServer host control domain; the remaining space is available for NetScaler VPX and other virtual machines.
Network Interface Card (NIC)	1 one gigabits per second (Gbps) NIC; 2 one Gbps NICs recommended

For information about installing XenServer, see the XenServer documentation at http://support.citrix.com/product/xens/.

The following table lists the virtual computing resources that XenServer must provide for each NetScaler nCore VPX.

Table 2. Minimum Virtual Computing Resources Required for Running NetScaler nCore VPX

Component	Requirement
Memory	2 GB
Virtual CPU (VCPU)	2
Virtual network interfaces	1

Note: For production use of NetScaler VPX, it is recommended that CPU priority (in virtual machine properties) be set to the highest level in order to improve scheduling behavior and network latency.

XenCenter System Requirements

XenCenter® is a Windows client application. It cannot run on the same machine as the XenServer® host. The following table describes the minimum system requirements.

Table 3. Minimum System Requirements for XenCenter Installation

Component	Requirement
Operating system	Windows XP, Windows Server 2003, or Windows Vista
.NET framework	Version 2.0 or later
CPU	750 megahertz (MHz) Recommended: 1 gigahertz (GHz) or faster
RAM	1 GB Recommended: 2 GB
Network Interface Card (NIC)	100 megabits per second (Mbps) or faster NIC

For information about installing XenCenter, see the XenServer documentation at http://support.citrix.com/product/xens/.

Installing NetScaler Virtual Appliances on XenServer by Using XenCenter

After you have installed and configured XenServer and XenCenter, you can use XenCenter to install virtual appliances on XenServer. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running XenServer.

After you have used XenCenter to install the initial NetScaler virtual appliance (.xva image) on XenServer, you have the option to use Command Center to provision NetScaler VPX. For more information, see the Command Center documentation.

To install NetScaler virtual appliances on XenServer by using XenCenter

1. Start XenCenter on your workstation.
2. On the **Server** menu, click **Add**.
3. In the **Add New Server** dialog box, in the **Hostname** text box, type the IP address or DNS name of the XenServer that you want to connect to.
4. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click **Connect**. The XenServer name appears in the navigation pane with a green circle, which indicates that the XenServer is connected.
5. In the navigation pane, click the name of the XenServer on which you want to install NetScaler VPX.
6. On the **VM** menu, click **Import**.
7. In the **Import** dialog box, in **Import file** name, browse to the location at which you saved the NetScaler VPX .xva image file. Make sure that the **Exported VM** option is selected, and then click **Next**.
8. Select the XenServer on which you want to install the virtual appliance, and then click **Next**.
9. Select the local storage repository in which to store the virtual appliance, and then click **Import** to begin the import process.
10. You can add, modify, or delete virtual network interfaces as required. When finished, click **Next**.
11. Click **Finish** to complete the import process.

    Note: To view the status of the import process, click the **Log** tab.
12. If you want to install another virtual appliance, repeat steps 5 through 11.

Installing NetScaler Virtual Appliances on VMware ESX

Before installing Citrix® NetScaler® virtual appliances on VMware ESX, make sure that VMware ESX server is installed on a machine with adequate system resources. To install virtual appliances on VMware ESX version 4.0, you use VMware vSphere client. On VMware ESX version 3.5, you use the VMware Open Virtualization Format (OVF) tool. The client or tool must be installed on a remote machine that can connect to VMware ESX through the network.

After the installation, you can use vSphere client 4.0 to manage virtual appliances on VMware ESX 4.0, or you can use VMware Infrastructure (VI) client 2.5 to manage virtual appliances on VMware ESX 3.5.

Note:

The VMware vSphere client shows the guest operating system as "Sun Solaris 10" for NetScaler VPX. This is by design because VMware ESX 3.5 does not recognize FreeBSD.

After the initial configuration of the NetScaler appliance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Prerequisites for Installing NetScaler Virtual Appliances on VMware

Before you begin installing a virtual appliance, do the following:

- Install VMware ESX version 3.5 or later on hardware that meets the minimum requirements.
- Install VMware Client on a management workstation that meets the minimum system requirements.
- Install VMware OVF Tool (required for VMware ESX version 3.5) on a management workstation that meets the minimum system requirements.
- Download the NetScaler VPX setup files.
- Label the physical network ports of VMware ESX.
- Obtain NetScaler VPX license files. For more information about NetScaler VPX licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx122426.

VMware ESX Hardware Requirements

The following table describes the minimum system requirements for VMware EXS servers running NetScaler nCore VPX.

Table 1. Minimum System Requirements for VMware ESX Servers Running NetScaler nCore VPX

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT or AMD-V) enabled **Note:** To run NetScaler VPX, hardware support for virtualization must be enabled on the VMware ESX host. Make sure that the BIOS option for virtualization support is not disabled. For more information, see your BIOS documentation.
RAM	3 GB

Disk space	Locally attached storage (PATA, SATA, SCSI) with 40 GB of disk space available
Network	1 one gigabits per second (Gbps) NIC; 2 one Gbps NICs recommended (The network interfaces should be E1000.)

For information about installing VMware ESX, see http://www.vmware.com/.

The following table lists the virtual computing resources that the VMware ESX server must provide for each NetScaler nCore VPX.

Table 2. Minimum Virtual Computing Resources Required for Running NetScaler nCore VPX

Component	Requirement
Memory	2 GB
Virtual CPU (VCPU)	2
Virtual network interfaces	1 **Note:** If the virtual appliance is installed on ESX 3.5 or ESXi 3.5, you can install a maximum of 4 virtual network interfaces. If the virtual appliance is installed on ESX 4.0, the maximum is 10.
Disk space	20 GB **Note:** This is in addition to any disk requirements for the hypervisor.

Note: For production use of NetScaler VPX, the full memory allocation must be reserved. CPU MHz should also be reserved at least equal to the MHz of one CPU core in the system.

VMware vSphere Client 4.0 System Requirements

VMware vSphere is a client application that can run on Windows and Linux operating systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 3. Minimum System Requirements for VMware vSphere Client Installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "vSphere Compatibility Matrixes" PDF file at http://kb.vmware.com/.
CPU	750 megahertz (MHz); 1 gigahertz (GHz) or faster recommended

RAM	1 GB; 2 GB recommended
Network Interface Card (NIC)	100 Mbps or faster NIC

For information about installing vSphere client 4.0, see http://www.vmware.com/.

Note: When you connect the vSphere client 4.0 to ESX 3.5, the vSphere client downgrades to VMware Infrastructure (VI) client version 2.5, which is the only version that is compatible with ESX 3.5.

OVF Tool 1.0 System Requirements

OVF Tool is a client application that can run on Windows and Linux systems. It cannot run on the same machine as the VMware ESX server. You need to use VMware OVF Tool version 1.0 for installing virtual appliances on ESX 3.5. The following table describes the minimum system requirements.

Table 4. Minimum System Requirements for OVF Tool Installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended.
RAM	1 GB Minimum, 2 GB recommended.
Network Interface Card (NIC)	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.

Downloading the NetScaler VPX Setup Files

The NetScaler VPX setup package for VMWare ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from MyCitrix.com. You will need a My Citrix account to log on. If you do not have a My Citrix account, access the home page at http://www.mycitrix.com, click the **New Users** link, and follow the instructions to create a new My Citrix account.

Once logged in, navigate the following path from the My Citrix home page:

MyCitrix.com > Downloads > NetScaler > Virtual Appliances.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-9.3-39.8-disk1.vmdk)

- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-9.3-39.8.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-9.3-39.8.mf)

Labeling the Physical Network Ports of VMware ESX

Before installing a NetScaler virtual appliance, you need to label at least one physical network port of VMware ESX in a particular format. The labeling format is NS_NIC_1_1, NS_NIC_1_2, and so on. These ports will be used by the virtual appliances that you install. An interface can be used by more than one virtual appliance.

To label the physical network ports of VMware ESX server

1. Log on to the VMware ESX server by using the vSphere client.
2. On the vSphere client, select the **Configuration** tab, and then click **Networking**.
3. At the top-right corner, click **Add Networking**, to start the **Add Network Wizard**.
4. In **Connection Type**, select **Virtual Machine**, and then click **Next**.
5. Scroll through the list of vSwitch physical adapters, and choose the physical port that will map to interface 1/1 on the virtual appliances.
6. Enter **NS_NIC_1_1** as the name of the vSwitch that will be associated with interface 1/1 of the virtual appliances.
7. Click **Next** to finish the vSwitch creation. Repeat the procedure, beginning with step 2, to add any additional interfaces to be used by your virtual appliances. Label the interfaces sequentially, in the correct format (for example, **NS_NIC_1_2**).

Installing NetScaler Virtual Appliances on VMware ESX 4.0

After you have installed and configured VMware ESX 4.0, you can use VMware vSphere client to install virtual appliances on the VMware ESX. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running VMware ESX.

To install NetScaler virtual appliances on VMware ESX 4.0 by using VMware vSphere Client

1. Start the VMware vSphere client on your workstation.
2. In the **IP address / Name** text box, type the IP address of the VMware ESX server that you want to connect to.
3. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click **Login**.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the NetScaler VPX setup files, select the .ovf file, and click **Next**.
6. Map the networks shown in the VPX OVF template to the networks that you configured on the ESX host. Click **Next** to start installing VPX on VMware ESX. When installation is complete, a pop-up window informs you of the successful installation.
7. You are now ready to start the NetScaler VPX. In the navigation pane, select the NetScaler VPX that you have just installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.
8. If you want to install another virtual appliance, repeat steps 4 through 6.

Installing NetScaler Virtual Appliances on VMware ESX 3.5

To install virtual appliances on ESX 3.5, you need to use the VMware OVF tool, version 1.0. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running VMware ESX. After installation, you can use the VMware Infrastructure (VI) client 2.5 to manage the virtual appliances on VMware ESX version 3.5.

Note: You cannot use version 4.0 of the vSphere client for installing virtual appliances on ESX 3.5. If you connect the vSphere 4.0 client to ESX 3.5, the vSphere client downgrades to VI client version 2.5, which supports only the OVF 0.9 standard. The NetScaler VPX installation package is based on the OVF 1.0.

To install NetScaler virtual appliances on VMware ESX 3.5 by using the VMware OVF Tool

1. On your workstation, open the command-line interface and execute the following command:

   ```
   ovftool <path of the NetScaler VPX OVF file>
   vi://<Username>:<Password>@<IP address of the ESX server>
   ```

   For example, in Windows command shell, type:

   ```
   ovftool c:/NetScalerVPX vi://root:free@<10.217.20.14>
   ```

2. When the OVF tool has installed the virtual appliances on the ESX server, use the VI client to log on to the VMware ESX server on which you performed the installation.

3. In the navigation pane, right-click a virtual appliance that you want to enable, and then click **Power On**. Repeat this for each virtual appliance you want to enable.

4. Click the **Console** tab to emulate a console port.

Installing Citrix NetScaler Virtual Appliances on Microsoft Server 2008 R2

To install Citrix® NetScaler® virtual appliances on Microsoft Windows Server 2008 R2, you must first install Windows Server 2008 R2, with the Hyper-V role enabled, on a machine with adequate system resources. While installing the Hyper-V role, make sure you specify the network interface cards (NICs) on the server that Hyper-V will use to create the virtual networks. You can reserve some NICs for the host. Use Hyper-V Manager to perform the NetScaler VPX installation.

NetScaler VPX for Hyper-V is delivered in virtual hard disk (VHD) format. It includes the default configuration for elements such as CPU, network interfaces, and hard-disk size and format. After you install NetScaler VPX, you can configure the network adapters on VPX, add virtual NICs, and then assign the NetScaler IP address, subnet mask, and gateway, and complete the basic configuration of the virtual appliance.

Note:

NetScaler VPX for Hyper-V does not support L2 mode.

After the initial configuration of the NetScaler appliance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Prerequisites for Installing NetScaler VPX on Microsoft Server 2008 R2

Before you begin installing a virtual appliance, do the following:

- Enable the Hyper-V role on Windows Server 2008 R2. For more information, see http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspx.
- Download the VPX setup files.
- Obtain NetScaler VPX license files. For more information about NetScaler VPX licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx122426.

Microsoft Server 2008 R2 Hardware Requirements

The following table describes the minimum system requirements for Microsoft Server 2008 R2.

Table 1. Minimum System Requirements for Microsoft Server 2008 R2

Component	Requirement
CPU	1.4 GHz 64-bit processor
RAM	3 GB
Disk Space	32 GB or greater

For more information about Windows Server 2008 R2 system requirements, see http://www.microsoft.com/windowsserver2008/en/us/system-requirements.aspx.

For information about installing Microsoft Server 2008 R2, see http://technet.microsoft.com/en-us/library/dd379511(WS.10).aspx.

The following table lists the virtual computing resources for each NetScaler nCore VPX.

Table 2. Minimum Virtual Computing Resources Required for Running NetScaler nCore VPX

Component	Requirement
RAM	2 GB
Virtual CPU	2
Disk Space	20 GB
Virtual Network Interfaces	1

Downloading the NetScaler VPX Setup Files

NetScaler VPX for Hyper-V is delivered in virtual hard disk (VHD) format. You can download the files from MyCitrix.com. You will need a My Citrix account to log on. If you do not have a My Citrix account, access the home page at http://www.mycitrix.com, click the **New Users** link, and follow the instructions to create a new My Citrix account.

To download the NetScaler VPX setup files

1. In a Web browser, go to http://www.citrix.com/ and click **My Citrix**.
2. Type your user name and password.
3. Click **Downloads**.
4. In **Search Downloads by Product**, select **NetScaler**.
5. Under **Virtual Appliances**, click **NetScaler VPX**.
6. Copy the compressed file to your server.

Installing NetScaler VPX on Microsoft Server 2008 R2

After you have enabled the Hyper-V role on Microsoft Server 2008 R2 and extracted the VPX files, you can use Hyper-V Manager to install NetScaler VPX. After you import the virtual machine, you need to configure the virtual NICs by associating them to the virtual networks created by Hyper-V.

You can configure a maximum of eight virtual NICs. Even if the physical NIC is DOWN, the virtual appliance assumes that the virtual NIC is UP, because it can still communicate with the other virtual appliances on the same host (server).

Note: You cannot change any settings while the virtual appliance is running. Shut down the virtual appliance and then make changes.

To install NetScaler VPX on Microsoft Server 2008 R2 by using Hyper-V Manager

1. To start Hyper-V Manager, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the navigation pane, under **Hyper-V Manager**, select the server on which you want to install NetScaler VPX.
3. On the **Action** menu, click **Import Virtual Machine**.
4. In the **Import Virtual Machine** dialog box, in **Location**, specify the path of the folder that contains the NetScaler VPX software files, and then select **Copy the virtual machine (create a new unique ID)**. This folder is the parent folder that contains the Snapshots, Virtual Hard Disks, and Virtual Machines folders.

   Note: If you received a compressed file, make sure that you extract the files into a folder before you specify the path to the folder.
5. Click **Import**.
6. Verify that the virtual appliance that you imported is listed under **Virtual Machines**.
7. To install another virtual appliance, repeat steps 2 through 6.

   Important: Make sure that you extract the files to a different folder in step 4.

To configure virtual NICs on the NetScaler VPX

1. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**.
2. In the **Settings for** <virtual appliance name> dialog box, click **Add Hardware** in the left pane.
3. In the right pane, from the list of devices, select **Network Adapter**.
4. Click **Add**.
5. Verify that **Network Adapter** (**not connected**) appears in the left pane.
6. Select the network adapter in the left pane.
7. In the right pane, from the **Network** drop-down list, select the virtual network to connect the adapter to.
8. To select the virtual network for additional network adapters that you want to use, repeat steps 6 and 7.
9. Click **Apply**, and then click **OK**.

To configure NetScaler VPX

1. Right-click the virtual appliance that you previously installed, and then select **Start**.
2. Access the console by double-clicking the virtual appliance.
3. Type the NetScaler IP address, subnet mask, and gateway for your virtual appliance.

You have completed the basic configuration of your virtual appliance. Type the IP address in a Web browser to access the virtual appliance.

Configuring the Basic System Settings

After installing a Citrix® NetScaler® VPX virtual appliance, you need to access it to configure the basic settings. Initially, you must access the NetScaler command line through the respective management application of the virtualization host (either Citrix XenCenter for Citrix XenServer or VMware vSphere client for VMware ESX) to specify a NetScaler IP (NSIP) address, subnet mask, and default gateway. The NSIP is the management address at which you can then access the NetScaler command line, through an SSH client, or access the configuration utility. You can use either of these access methods, or the console, to continue with basic configuration.

To access the configuration utility, type the NSIP into the address field of any browser (for example, `http://<NSIP_address>`). You need Java RunTime Environment (JRE) version 1.6 or later.

Setting Up the Initial Configuration by Using the NetScaler VPX Console

Your first task after installing a NetScaler virtual appliance on a virtualization host is to use the NetScaler VPX console in the XenCenter client or vSphere client to configure the following initial settings.

Note: If you have installed a virtual appliance on XenServer by using Command Center, you do not have to configure these settings. Command Center implicitly configures the settings during installation. For more information about provisioning VPX from Command Center, see the Command Center documentation.

NetScaler IP address (NSIP):

The IP address at which you access a NetScaler or a NetScaler virtual appliance for management purposes. A physical NetScaler or virtual appliance can have only one NSIP. You must specify this IP address when you configure the virtual appliance for the first time. You cannot remove an NSIP address.

Netmask:

The subnet mask associated with the NSIP address.

Default Gateway:

You must add a default gateway on the virtual appliance if you want access it through SSH or the configuration utility from an administrative workstation or laptop that is on a different network.

To configure the initial settings on the virtual appliance through the VPX Console by using the management application

1. Connect to the XenServer or VMware ESX server on which the virtual appliance is installed by using XenCenter or vSphere client, respectively.
2. In the details pane, on the **Console** tab, log on to the virtual appliance by using the administrator credentials.
3. At the prompts, enter the NSIP address, subnet mask, and default gateway, and then save the configuration.

After you have set up an initial configuration through the NetScaler VPX Console in the management application, you can use either the NetScaler command-line interface or the configuration utility to complete the configuration or to change the initial settings.

Configuring NetScaler VPX by Using the Command-Line Interface

You can use the NetScaler command-line to set up the NSIP, Mapped IP (MIP), Subnet IP (SNIP), and hostname. You can also configure advanced network settings and change the time zone.

For information about MIP, SNIP, other NetScaler-owned IP addresses, and network settings, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

To complete initial configuration by using the NetScaler command line

1. Use either the SSH client or the NetScaler VPX Console in XenCenter to access the NetScaler command line.
2. Log on to the virtual appliance, using the administrator credentials.
3. At the NetScaler command prompt, type `config ns` to run the configuration script.
4. To complete the initial configuration, follow the prompts.

You have now completed the basic configuration of the virtual appliance. To continue the configuration process, choose one of the following options:

Citrix NetScaler Load Balancing Switch.

If you are configuring the virtual appliance as a standard NetScaler load balancing switch with other licensed features, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Citrix Application Firewall.

If you are configuring the virtual appliance as a standalone application firewall, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

For more information about the various features supported on NetScaler VPX, see Features at a Glance.

Configuring NetScaler VPX by Using the Configuration Utility

To use the Setup Wizard to set up the NetScaler virtual appliance, you must access the configuration utility from your Web browser. You can use the Setup Wizard to configure the NSIP, MIP, SNIP, hostname, and default gateway. You can also configure settings for a Web application by using an application template. You can also configure the NetScaler VPX as a load balancer for Citrix XenDesktop® or Citrix XenApp™.

For information about MIP, SNIP, and network settings, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

For information about application templates, see the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

For information about the load balancing feature of a virtual appliance, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To configure initial settings by using the configuration utility

1. In the address field of a Web browser, type: `http://<NSIP address>`
2. In **User Name** and **Password**, type the administrator credentials.
3. In **Start in**, select **Configuration**, and then click **Login**.
4. In the **Setup Wizard**, click **Next** and follow the instructions.

You have now completed the basic configuration of the virtual appliance. To continue the configuration process, choose one of the following options:

Citrix NetScaler Load Balancing Switch.

If you are configuring the virtual appliance as a standard NetScaler load balancing switch with other licensed features, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Citrix Application Firewall.

If you are configuring the virtual appliance as a standalone application firewall, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

For more information about the various features supported by NetScaler VPX, see Features at a Glance.

Understanding Common Network Topologies

As described in Physical Deployment Modes, you can deploy the Citrix® NetScaler® appliance either inline between the clients and servers or in one-arm mode. Inline mode uses a two-arm topology, which is the most common type of deployment.

Setting Up Common Two-Arm Topologies

In a two-arm topology, one network interface is connected to the client network and another network interface is connected to the server network, ensuring that all traffic flows through the NetScaler. This topology might require you to reconnect your hardware and also might result in a momentary downtime. The basic variations of two-arm topology are multiple subnets, typically with the NetScaler on a public subnet and the servers on a private subnet, and transparent mode, with both the NetScaler and the servers on the public network.

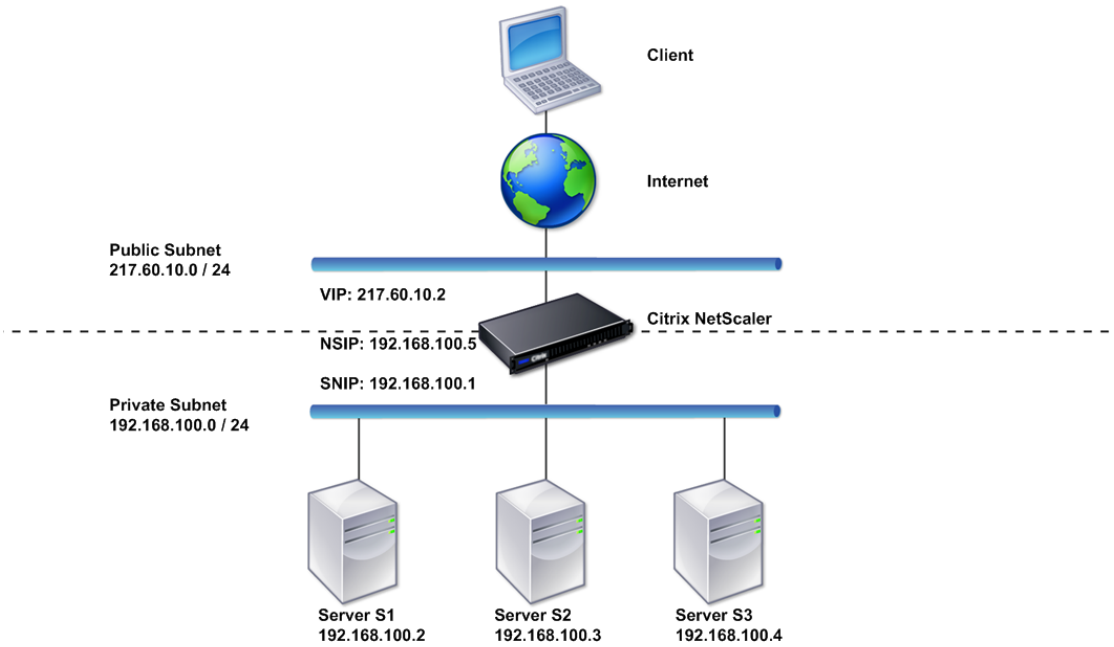
Setting Up a Simple Two-Arm Multiple Subnet Topology

One of the most commonly used topologies has the NetScaler inline between the clients and the servers, with a vserver configured to handle the client requests. This configuration is used when the clients and servers reside on different subnets. In most cases, the clients and servers reside on public and private subnets, respectively.

For example, consider a NetScaler deployed in two-arm mode for managing servers S1, S2, and S3, with a vserver of type HTTP configured on the NetScaler, and with HTTP services running on the servers. The servers are on a private subnet and a SNIP is configured on the NetScaler to communicate with the servers. The Use SNIP (USNIP) option must be enabled on the NetScaler so that it uses the SNIP instead of the MIP.

As shown in the following figure, the VIP and a SNIP are on public subnet 217.60.10.0, and the NSIP, the servers, and another SNIP are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for Two-Arm Mode, Multiple Subnets



Task overview: To deploy a NetScaler in two-arm mode with multiple subnets

1. Configure the NSIP and default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.

2. Configure the SNIP, as described in the Configuring Subnet IP Addresses (SNIPs).

3. Enable the USNIP option, as described in the To enable or disable USNIP mode by using the NetScaler command line.

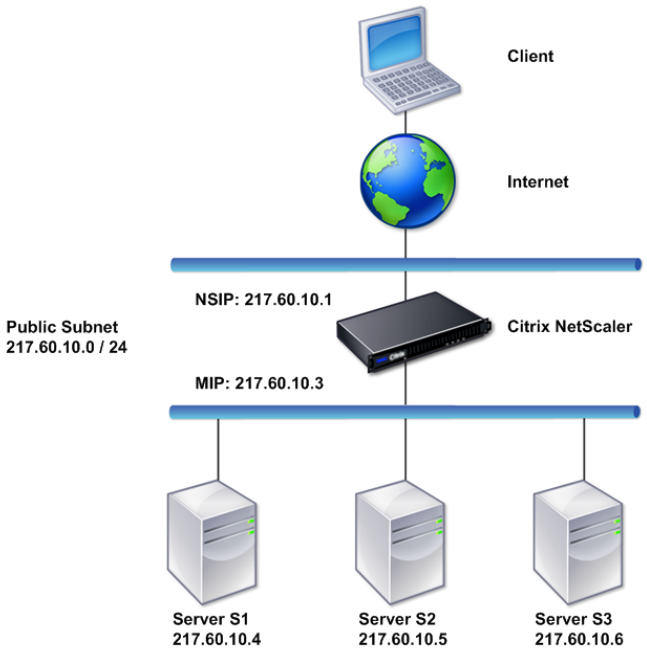
4. Configure the vserver and the services, as described in the Creating a Virtual Server and Configuring Services.

5. Connect one of the network interfaces to a private subnet and the other interface to a public subnet.

Setting Up a Simple Two-Arm Transparent Topology

Use transparent mode if the clients need to access the servers directly, with no intervening vserver. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a NetScaler is placed between the client and the server, so the traffic must pass through the NetScaler. You must enable L2 mode for bridging the packets. The NSIP and MIP are on the same public subnet, 217.60.10.0/24.

Figure 1. Topology Diagram for Two-Arm, Transparent Mode



Task overview: To deploy a NetScaler in two-arm, transparent mode

1. Configure the NSIP, MIP, and default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.
2. Enable L2 mode, as described in the Enabling and Disabling Layer 2 Mode.
3. Configure the default gateway of the managed servers as the MIP.
4. Connect the network interfaces to the appropriate ports on the switch.

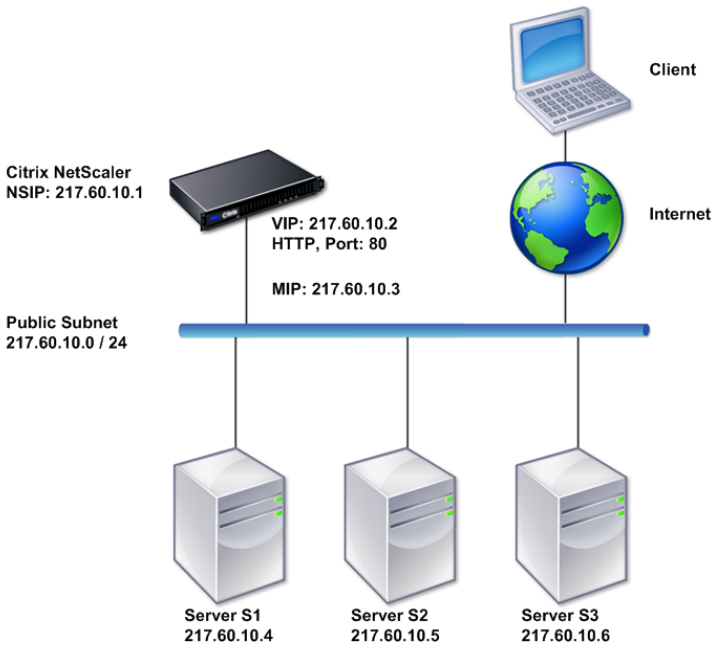
Setting Up Common One-Arm Topologies

The two basic variations of one-arm topology are with a single subnet and with multiple subnets.

Setting Up a Simple One-Arm Single Subnet Topology

You can use a one-arm topology with a single subnet when the clients and servers reside on the same subnet. For example, consider a NetScaler deployed in one-arm mode for managing servers S1, S2, and S3. A vserver of type HTTP is configured on a NetScaler, and HTTP services are running on the servers. As shown in the following figure, the NetScaler IP address (NSIP), the Mapped IP address (MIP), and the server IP addresses are on the same public subnet, 217.60.10.0/24.

Figure 1. Topology Diagram for One-Arm Mode, Single Subnet



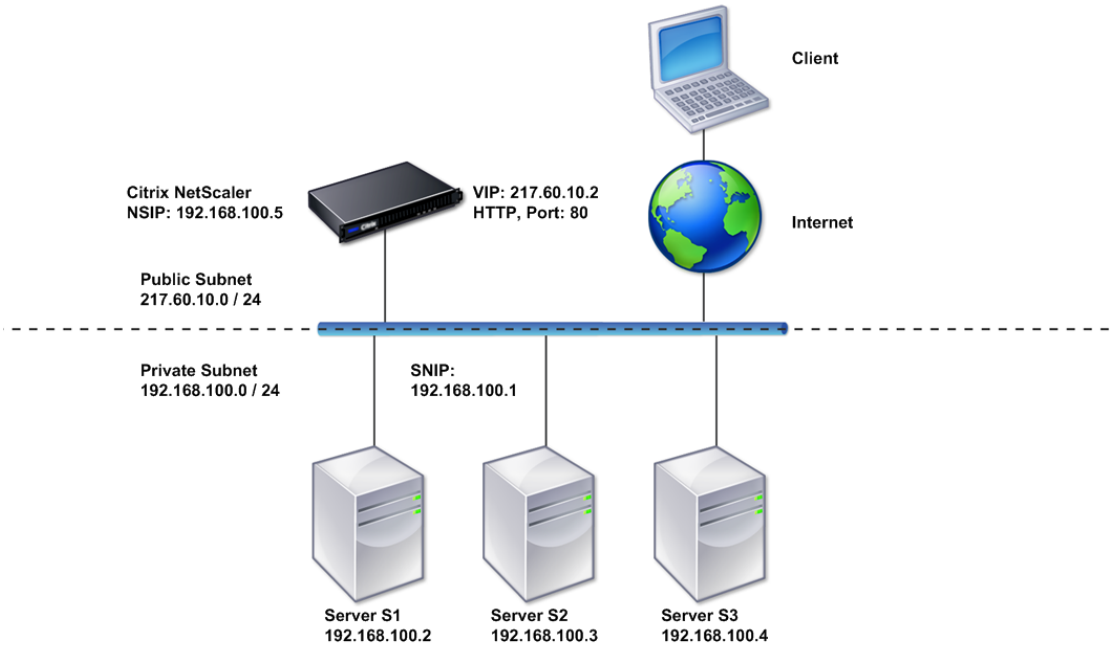
Task overview: To deploy a NetScaler in one-arm modewith a single subnet

1. Configure the NSIP, MIP, and the default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.
2. Configure the vserver and the services, as described in Creating a Virtual Server and Configuring Services.
3. Connect one of the network interfaces to the switch.

Setting Up a Simple One-Arm Multiple Subnet Topology

You can use a one-arm topology with multiple subnets when the clients and servers reside on the different subnets. For example, consider a NetScaler deployed in one-arm mode for managing servers S1, S2, and S3, with the servers connected to switch SW1 on the network. A vserver of type HTTP is configured on the NetScaler, and HTTP services are running on the servers. These three servers are on the private subnet, so a subnet IP address (SNIP) is configured to communicate with them. The Use Subnet IP address (USNIP) option must be enabled so that the NetScaler uses the SNIP instead of a MIP. As shown in the following figure, the virtual IP address (VIP) is on public subnet 217.60.10.0/24; the NSIP, SNIP, and the server IP addresses are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for One-Arm Mode, Multiple Subnets



Task overview: To deploy a NetScaler in one-arm mode with multiple subnets

1. Configure the NSIP and the default gateway, as described in Configuring a NetScaler by Using the Command Line Interface.

2. Configure the SNIP and enable the USNIP option, as described in the Configuring Subnet IP Addresses (SNIPs).

3. Configure the vserver and the services, as described in Creating a Virtual Server and Configuring Services.

4. Connect one of the network interfaces to the switch.

Configuring System Management Settings

Once your initial configuration is in place, you can configure settings to define the behavior of the Citrix® NetScaler® appliance and facilitate connection management. You have a number of options for handling HTTP requests and responses. Routing, bridging, and MAC based forwarding modes are available for handling packets not addressed to the NetScaler. You can define the characteristics of your network interfaces and can aggregate the interfaces. To prevent timing problems, you can synchronize the NetScaler clock with a Network Time Protocol (NTP) server. The NetScaler can operate in various DNS modes, including as an authoritative domain name server (ADNS). You can set up SNMP for system management and customize syslog logging of system events. Before deployment, verify that your configuration is complete and correct.

Configuring System Settings

Configuration of system settings includes basic tasks such as configuring HTTP ports to enable connection keep-alive and server offload, setting the maximum number of connections for each server, and setting the maximum number of requests per connection. You can enable client IP address insertion for situations in which a proxy IP address is not suitable, and you can change the HTTP cookie version.

You can also configure a NetScaler to open FTP connections on a controlled range of ports instead of ephemeral ports for data connections. This improves security, because opening all ports on the firewall is insecure. You can set the range anywhere from 1,024 to 64,000.

Before deployment, go through the verification checklists to verify your configuration. To configure HTTP parameters and the FTP port range, use the NetScaler configuration utility.

You can modify the types of HTTP parameters described in the following table.

Table 1. HTTP Parameters

Parameter Type	Specifies
HTTP Port Information	The Web server HTTP ports used by your managed servers. If you specify the ports, the NetScaler can perform request switching for any client request that has a destination port matching a specified port. **Note:** If an incoming client request is not destined for a service or a virtual server that is specifically configured on the NetScaler, the destination port in the request must match one of the globally configured HTTP ports. This allows the NetScaler to perform connection keep-alive and server off-load.

Limits	The maximum number of connections to each managed server, and the maximum number of requests sent over each connection. For example, if set **Max Connections** to 500, and the NetScaler is managing three servers, it can open a maximum of 500 connections to each of the three servers. By default, the NetScaler can create an unlimited number of connections to any of the servers it manages. To specify an unlimited number of requests per connection, set **Max Requests** to 0. **Note:** If you are using the Apache HTTP server, you must set **Max Connections** equal to the value of the MaxClients parameter in the Apache httpd.conf file. Setting this parameter is optional for other web servers.
Client IP Insertion	Enable/disable insertion of the client's IP address into the HTTP request header. You can specify a name for the header field in the adjacent text box. When a Web server managed by a NetScaler receives a mapped IP address or a subnet IP address, the server identifies it as the client's IP address. Some applications need the client's IP address for logging purposes or to dynamically determine the content to be served by the web server. You can enable insertion of the actual client IP address into the HTTP header request sent from the client to one, some, or all servers managed by the NetScaler. You can then access the inserted address through a minor modification to the server (using an Apache module, ISAPI interface, or NSAPI interface).
Cookie Version	The HTTP cookie version to use when COOKIEINSERT persistence is configured on a virtual server. The default, version 0, is the most common type on the Internet. Alternatively, you can specify version 1.
Requests/Responses	Options for handling certain types of requests, and enable/disable logging of HTTP error responses.
Server Header Insertion	Insert a server header in NetScaler-generated HTTP responses.

To configure HTTP parameters by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change HTTP parameters**.
3. In the **Configure HTTP parameters** dialog box, specify values for some or all of the parameters that appear under the headings listed in the table above.
4. Click **OK**.

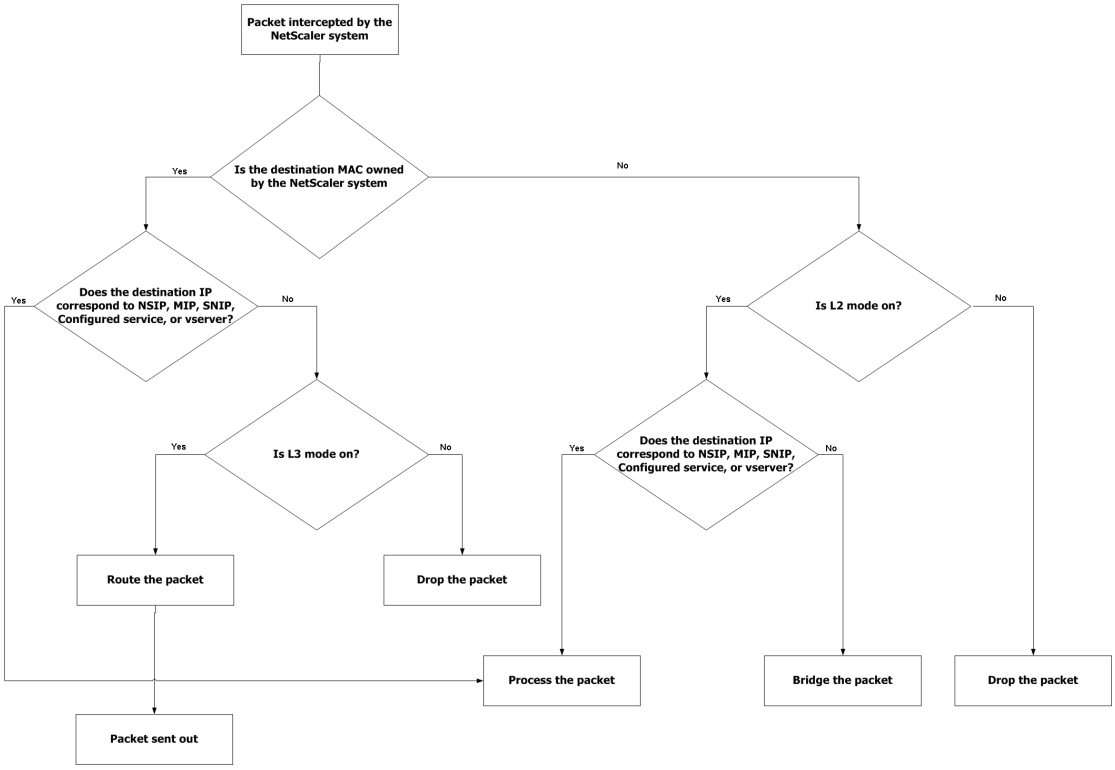
To set the FTP port range by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change global system settings**.
3. Under **FTP Port Range**, in the **Start Port** and **End Port** text boxes, type the lowest and highest port numbers, respectively, for the range you want to specify (for example, 5000 and 6000).
4. Click **OK**.

Configuring Modes of Packet Forwarding

The NetScaler can either route or bridge packets that are not destined for an IP address owned by the NetScaler (that is, the IP address is not the NSIP, a MIP, a SNIP, a configured service, or a configured vserver). By default, L3 mode (routing) is enabled and L2 mode (bridging) is disabled, but you can change the configuration. The following flow chart shows how the NetScaler evaluates packets and either processes, routes, bridges, or drops them.

Figure 1. Interaction between Layer 2 and Layer 3 Modes



A NetScaler can use the following modes to forward the packets it receives:

- Layer 2 (L2) Mode
- Layer 3 (L3) Mode
- MAC-Based Forwarding Mode

Enabling and Disabling Layer 2 Mode

Layer 2 mode controls the Layer 2 forwarding (bridging) function. You can use this mode to configure a NetScaler to behave as a Layer 2 device and bridge the packets that are not destined for it. When this mode is enabled, packets are not forwarded to any of the MAC addresses, because the packets can arrive on any interface of the NetScaler and each interface has its own MAC address.

With Layer 2 mode disabled (which is the default), a NetScaler drops packets that are not destined for one of its MAC address. If another Layer 2 device is installed in parallel with a NetScaler, Layer 2 mode must be disabled to prevent bridging (Layer 2) loops. You can use the configuration utility or the command line to enable Layer 2 mode.

Note: The NetScaler does not support spanning tree protocol. To avoid loops, if you enable L2 mode, do not connect two interfaces on the NetScaler to the same broadcast domain.

To enable or disable Layer 2 mode by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable/disable Layer 2 mode and verify that it has been enabled/disabled:

- `enable ns mode <Mode>`
- `disable ns mode <Mode>`
- `show ns mode`

Examples

```
> enable ns mode l2
 Done
> show ns mode

      Mode                  Acronym          Status
      -------               -------          ------
 1)    Fast Ramp               FR              ON
 2)    Layer 2 mode             L2              ON
 .
 .
 .
 Done
>

> disable ns mode l2
```

```
 Done
> show ns mode

      Mode                  Acronym          Status
      -------               -------          ------
1)    Fast Ramp               FR               ON
2)    Layer 2 mode            L2               OFF
.
.
.
 Done
>
```

To enable or disable Layer 2 mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change modes**.
3. In the **Configure Modes** dialog box, to enable Layer 2 mode, select the **Layer 2 Mode** check box. To disable Layer 2 mode, clear the check box.
4. Click **OK**. The **Enable/Disable Mode(s)?** message appears in the details pane.
5. Click **Yes**.

Enabling and Disabling Layer 3 Mode

Layer 3 mode controls the Layer 3 forwarding function. You can use this mode to configure a NetScaler to look at its routing table and forward packets that are not destined for it. With Layer 3 mode enabled (which is the default), a NetScaler performs route table lookups and forwards all packets that are not destined for any NetScaler-owned IP address. If you disable Layer 3 mode, the NetScaler drops these packets.

To enable or disable Layer 3 mode by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable/disable Layer 3 mode and verify that it has been enabled/disabled:

- `enable ns mode <Mode>`
- `disable ns mode <Mode>`
- `show ns mode`

Examples

```
> enable ns mode l3
 Done
> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
 1)   Fast Ramp                 FR               ON
 2)   Layer 2 mode               L2               OFF
 .
 .
 .
 9)   Layer 3 mode (ip forwarding)   L3               ON
 .
 .
 .
 Done
>

> disable ns mode l3
 Done
> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
```

```
1)    Fast Ramp                  FR              ON
2)    Layer 2 mode                L2              OFF
.
.
.
9)    Layer 3 mode (ip forwarding)  L3              OFF
.
.
.
Done
>
```

To enable or disable Layer 3 mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change modes**.
3. In the **Configure Modes** dialog box, to enable Layer 3 mode, select the **Layer 3 Mode (IP Forwarding)** check box. To disable Layer 3 mode, clear the check box.
4. Click **OK**. The **Enable/Disable Mode(s)?** message appears in the details pane.
5. Click **Yes**.

Enabling and Disabling MAC-Based Forwarding Mode

You can use MAC-based forwarding to process traffic more efficiently and avoid multiple-route or ARP lookups when forwarding packets, because the NetScaler remembers the MAC address of the source. To avoid multiple lookups, the NetScaler caches the source MAC address of every connection for which it performs an ARP lookup, and it returns the data to the same MAC address.

MAC-based forwarding is useful when you use VPN devices because the NetScaler ensures that all traffic flowing through a particular VPN passes through the same VPN device.

The following figure shows the process of MAC-based forwarding.

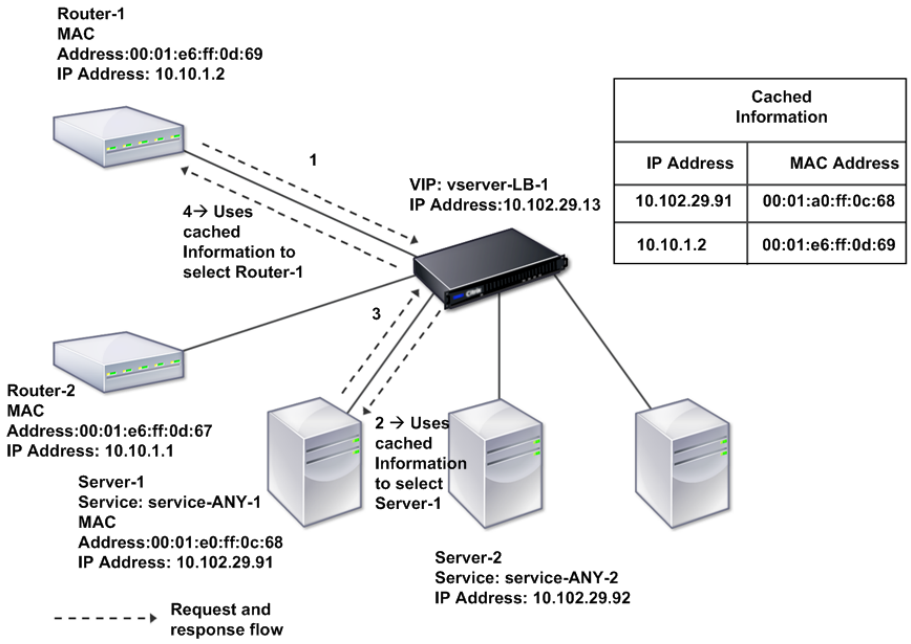


Figure 1. MAC-Based Forwarding Process

When MAC-based forwarding is enabled, a NetScaler caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server responds through a NetScaler, the NetScaler sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when a NetScaler initiates a connection, it uses the route and ARP tables for the lookup function. To enable MAC-based forwarding, use the configuration utility or the command line.

Some deployments require the incoming and outgoing paths to flow through different routers. In these situations, MAC-based forwarding breaks the topology design. For a global server load balancing (GSLB) site that requires the incoming and outgoing paths to flow through different routers, you must disable MAC-based forwarding and use the NetScaler unit's default router as the outgoing router.

With MAC-based forwarding disabled and Layer 2 or Layer 3 connectivity enabled, a route table can specify separate routers for outgoing and incoming connections. To disable MAC-based forwarding, use the configuration utility or the command line.

To enable or disable MAC-based forwarding by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable/disable MAC-based forwarding mode and verify that it has been enabled/disabled:

- `enable ns mode <Mode>`
- `disable ns mode <Mode>`
- `show ns mode`

Example

```
> enable ns mode mbf
 Done
> show ns mode

      Mode                  Acronym          Status
      -------               -------          ------
 1)   Fast Ramp               FR               ON
 2)   Layer 2 mode             L2               OFF
 .
 .
 .
 6)   MAC-based forwarding      MBF              ON
 .
 .
 .
 Done
>

> disable ns mode mbf
 Done
```

```
> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
 1)   Fast Ramp                 FR               ON
 2)   Layer 2 mode               L2               OFF
 .
 .
 .
 6)   MAC-based forwarding       MBF              OFF
 .
 .
 .
 Done
>
```

To enable or disable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features** group, click **Change modes**.
3. In the **Configure Modes** dialog box, to enable MAC-based forwarding mode, select the **MAC Based Forwarding** check box. To disable MAC-based forwarding mode, clear the check box.
4. Click **OK**. The **Enable/Disable Mode(s)?** message appears in the details pane.
5. Click **Yes**.

Configuring Clock Synchronization

You can configure your NetScaler to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network. NTP uses User Datagram Protocol (UDP) port 123 as its transport layer. You have to add NTP servers in the NTP configuration file so that the NetScaler periodically gets updates from these servers.

If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site at http://www.ntp.org/.

To configure clock synchronization on your NetScaler

1. Log on to the NetScaler command line and enter the `shell` command.

2. At the shell prompt, copy the ntp.conf file from the /etc directory to the /nsconfig directory. If the file already exists in the /nsconfig directory, make sure that you remove the following entries from the ntp.conf file:

   ```
   restrict localhost

   restrict 127.0.0.2
   ```

   These entries are required only if you want to run the device as a time server. However, this feature is not supported on the NetScaler.

3. Edit /nsconfig/ntp.conf by typing the IP address for the desired NTP server under the file's server and restrict entries.

4. Create a file named rc.netscaler in the /nsconfig directory, if the file does not already exist in the directory.

5. Edit /nsconfig/rc.netscaler by adding the following entry: `/usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &`

   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   Note: If the time difference between the NetScaler and the time server is more than 1000 sec, the ntpd service terminates with a message to the NetScaler log. To avoid this, you need to start ntpd with the **-g** option,which forcibly syncs the time. Add the following entry in /nsconfig/rc.netscaler:

   ```
   /usr/sbin/ntpd -g -c /nsconfig/ntp.conf -l /var/log/ntpd.log &
   ```

   If you do not want to forcibly sync the time when there is a large difference, you can set the date manually and then start ntpd again. You can check the time difference between the NetScaler and the time server by executing the following command in the shell:

   ntpdate -q <IP address or domain name of the NTP server>

6. Reboot the NetScaler to enable clock synchronization.

   Note: If you want to start time synchronization before you restart the NetScaler, you can enter the

   /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ ntpd.log &

   command (which you added to the rc.netscaler file in step 5) at the shell prompt.

Configuring DNS

You can configure a NetScaler to function as an Authoritative Domain Name Server (ADNS), DNS proxy server, End Resolver, or Forwarder. You can add DNS resource records such as SRV Records, AAAA Records, A Records, MX Records, NS Records, CNAME Records, PTR Records, and SOA Records. Also, the NetScaler can balance the load on external DNS servers.

A common practice is to configure a NetScaler as a forwarder. For this configuration, you need to add external name servers. After you have added the external servers, you should verify that your configuration is correct.

For other configurations, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing vserver as the name server.

When adding name servers, you can specify IP addresses or virtual IP addresses (VIPs). If you use IP addresses, the NetScaler load balances requests to the configured name servers in a round robin manner. If you use VIPs, you can specify any load balancing method. For information about using a VIP, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To add a name server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a name server and verify the configuration:

- `add dns nameServer <IP>`
- `show dns nameServer <IP>`

  Example

```
> add dns nameServer 10.102.29.10
 Done
> show dns nameServer 10.102.29.10
1)      10.102.29.10  -  State: DOWN
 Done
>
```

To add a name server by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Name Servers**.
2. In the details pane, click **Add**.
3. In the **Create Name Server** dialog box, select **IP Address**.
4. In the **IP Address** text box, type the IP address of the name server (for example, `10.102.29.10`). If you are adding an external name server, clear the **Local** check box.
5. Click **Create**, and then click **Close**.
6. Verify that the name server you added appears in the **Name Servers** pane.

Configuring SNMP

The Simple Network Management Protocol (SNMP) network management application, running on a external computer, queries the SNMP agent on the NetScaler. The agent searches the management information base (MIB) for data requested by the network management application and sends the data to the application.

SNMP monitoring uses traps messages and alarms. SNMP traps messages are asynchronous events that the agent generates to signal abnormal conditions, which are indicated by alarms. For example, if you want to be informed when CPU utilization is above 90 percent, you can set up an alarm for that condition. The following figure shows a network with a NetScaler that has SNMP enabled and configured.

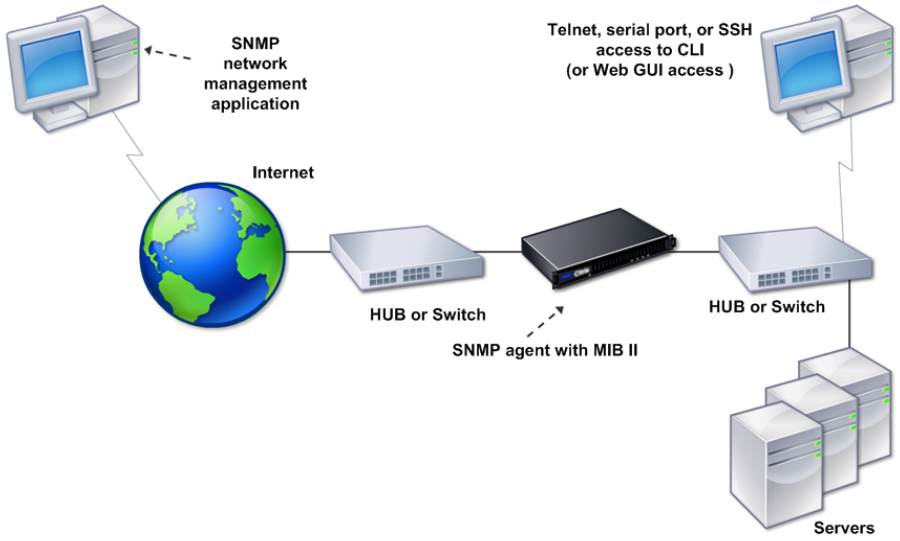


Figure 1. SNMP on the NetScaler

The SNMP agent on a NetScaler supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3). Because it operates in bilingual mode, the agent can handle SNMPv2 queries, such as Get-Bulk, and SNMPv1 queries. The SNMP agent also sends traps compliant with SNMPv2 and supports SNMPv2 data types, such as counter64. SNMPv1 managers (programs on other servers that request SNMP information from the NetScaler) use the NS-MIB-smiv1.mib file when processing SNMP queries. SNMPv2 managers use the NS-MIB-smiv2.mib file.

The NetScaler supports the following enterprise-specific MIBs:

A subset of standard MIB-2 groups

Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.

A system enterprise MIB

Provides system-specific configuration and statistics.

To configure SNMP, you specify which managers can query the SNMP agent, add SNMP trap listeners that will receive the SNMP trap messages, and configure SNMP Alarms.

Adding SNMP Managers

You can configure a workstation running a management application that complies with SNMP version 1, 2, or 3 to access a NetScaler. Such a workstation is called an SNMP manager. If you do not specify an SNMP manager on the NetScaler, the NetScaler accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the NetScaler accepts and responds to SNMP queries from only those specific IP addresses. When specifying the IP address of an SNMP manager, you can use the netmask parameter to grant access from entire subnets. You can add a maximum of 100 SNMP managers or networks.

To add an SNMP manager by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an SNMP manager and verify the configuration:

- `add snmp manager <IPAddress> ... [-netmask <netmask>]`
- `show snmp manager <IPAddress>`

  Example

```
> add snmp manager 10.102.29.5 -netmask 255.255.255.255
 Done
> show snmp manager 10.102.29.5
1)     10.102.29.5       255.255.255.255
 Done
>
```

To add an SNMP manager by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Managers**.
2. In the details pane, click **Add**.
3. In the **Add SNMP Manager** dialog box, in the **IP Address** text box, type the IP address of the workstation running the management application (for example, `10.102.29.5`).
4. Click **Create**, and then click **Close**.
5. Verify that the SNMP manager you added appears in the **Details** section at the bottom of the pane.

Adding SNMP Traps Listeners

After configuring the alarms, you need to specify the trap listener to which the NetScaler will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

To add an SNMP trap listener by using the NetScaler command line

At the NetScaler command prompt, type the following command to add an SNMP trap and verify that it has been added:

- `add snmp trap specific <IP>`
- `show snmp trap`

Example

```
> add snmp trap specific 10.102.29.3
 Done
> show snmp trap
Type       DestinationIP    DestinationPort  Version    SourceIP       Min-Severity   Community
----      -------------   ---------------  -------    --------      ------------  ---------
generic    10.102.29.9     162           V2       NetScaler IP    N/A          public
generic    10.102.29.5     162           V2       NetScaler IP    N/A          public
generic    10.102.120.101   162           V2       NetScaler IP    N/A          public
.
.
.
specific   10.102.29.3     162           V2       NetScaler IP    -           public
 Done
>
```

To add an SNMP trap listener by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Traps**.
2. In the details pane, click **Add**.
3. In the **Add SNMP Trap Destination** dialog box, in the **Destination IP Address** text box, type the IP address (for example, `10.102.29.3`).
4. Click **Create** and then click **Close**.
5. Verify that the SNMP trap you added appears in the **Details** section at the bottom of the pane.

Configuring SNMP Alarms

You configure alarms so that the NetScaler generates a trap message when an event corresponding to one of the alarms occurs. Configuring an alarm consists of enabling the alarm and setting the severity level at which a trap is generated. There are five severity levels: Critical, Major, Minor, Warning, and Informational. A trap is sent only when the severity of the alarm matches the severity specified for the trap.

Some alarms are enabled by default. If you disable an SNMP alarm, the NetScaler will not generate trap messages when corresponding events occur. For example, if you disable the Login-Failure SNMP alarm, the NetScaler will not generate a trap message when a login failure occurs.

To enable or disable an alarm by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable or disable an alarm and verify that it has been enabled or disabled:

- `set snmp alarm <trapName> [-state ENABLED | DISABLED ]`
- `show snmp alarm <trapName>`

Example

```
> set snmp alarm LOGIN-FAILURE -state ENABLED
 Done
> show snmp alarm LOGIN-FAILURE
   Alarm                  Alarm Threshold    Normal Threshold  Time  State     Severity     Logging
   -----                  ---------------    ----------------  ----  --------  ------------ --------
 1) LOGIN-FAILURE              N/A              N/A             N/A   ENABLED   -            ENABLED
 Done
>
```

To set the severity of the alarm by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the severity of the alarm and verify that the severity has been set correctly:

- `set snmp alarm <trapName> [-severity <severity>]`
- `show snmp alarm <trapName>`

Example

```
> set snmp alarm LOGIN-FAILURE -severity Major
 Done
> show snmp alarm LOGIN-FAILURE
   Alarm                    Alarm Threshold    Normal Threshold  Time  State     Severity     Logging
   -----                   ---------------   ----------------  ----  --------  ------------ --------
 1) LOGIN-FAILURE               N/A              N/A             N/A   ENABLED    Major       ENABLED
 Done
>
```

To configure alarms by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Alarms**.
2. In the details pane, select an alarm (for example, **LOGIN-FAILURE**), and then click **Open**.
3. In the **Configure SNMP Alarm** dialog box, to enable the alarm, select the **Enable** check box. To disable the alarm, clear the **Enable** check box.
4. In the **Severity** drop-down list, select a severity option (for example, **Major**).
5. Click **OK**, and then click **Close**.
6. Verify that the parameters for the SNMP alarm you configured are correctly configured by viewing the **Details** section at the bottom of the pane.

Configuring Syslog

You can customize logging of NetScaler and Access Gateway Enterprise Edition access events for the needs of your site. You can direct these logs either to files on the NetScaler or to external log servers. The NetScaler uses the Audit Server Logging feature for logging the states and status information collected by different modules in the kernel and by user-level daemons.

Syslog is used to monitor a NetScaler and to log connections, statistics, and so on. You can customize the two logging functions for system events messaging and syslog. The NetScaler internal event message generator passes log entries to the syslog server. The syslog server accepts these log entries and logs them. For more information about the Audit Server Logging feature, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Verifying the Configuration

After you finish configuring your system, complete the following checklists to verify your configuration.

Configuration Checklist

- The build running is:
- There are no incompatibility issues. (Incompatibility issues are documented in the build's release notes.)
- The port settings (speed, duplex, flow control, monitoring) are the same as the switch's port.
- Enough mapped IP addresses have been configured to support all server-side connections during peak times.
  - The number of configured mapped IP addresses is: ____
  - The expected number of simultaneous server connections is:

    [ ] 62,000 [ ] 124,000 [ ] Other____

Topology Configuration Checklist

The routes have been used to resolve servers on other subnets.

The routes entered are:

_____

- If the NetScaler is in a public-private topology, reverse NAT has been configured.
- The failover (high availability) settings configured on the NetScaler resolve in a one arm or two-arm configuration. All unused network interfaces have been disabled:
  _____
  _____
- If the NetScaler is placed behind an external load balancer, then the load balancing policy on the external load balancer is not "least connection."

  The load balancing policy configured on the external load balancer is:
  _____
- If the NetScaler is placed in front of a firewall, the session time-out on the firewall is set to a value greater than or equal to 300 seconds.

  The value configured for the session time-out is: _____

Server Configuration Checklist

- "Keep-alive" has been enabled on all the servers.

  The value configured for the keep-alive time-out is: _____

- The default gateway has been set to the correct value. (The default gateway should either be a NetScaler or upstream router.) The default gateway is: _____

- The server port settings (speed, duplex, flow control, monitoring) are the same as the switch port settings.

  _____

- If the Microsoft® Internet Information Server is used, buffering is enabled on the server.

- If an Apache Server is used, the MaxConn (maximum number of connections) parameter is configured on the server and on the NetScaler.

  The MaxConn (maximum number of connections) value that has been set is: _____

- If a NetScape® Enterprise Server™ is used, the maximum requests per connection parameter is set on the NetScaler.

  The maximum requests per connection value that has been set is: _____

Software Features Configuration Checklist

- Does the Layer 2 mode feature need to be disabled? (Disable if another Layer 2 device is working in para

  Reason for enabling or disabling:

  _____

- Does the MAC-based forwarding feature need to be disabled? (If the MAC address used by return traffic is

  Reason for enabling or disabling:

  _____

- Does host-based reuse need to be disabled? (Is there virtual hosting on the servers?)

  Reason for enabling or disabling:

  _____

- Do the default settings of the surge protection feature need to be changed?

  Reason for changing or not changing:

  _____

Access Checklist

- The system IPs can be pinged from the client-side network.

- The system IPs can be pinged from the server-side network.
- The managed server(s) can be pinged through the NetScaler.
- Internet hosts can be pinged from the managed servers.
- The managed server(s) can be accessed through the browser.
- The Internet can be accessed from managed server(s) using the browser.
- The system can be accessed using SSH.
- Admin access to all managed server(s) is working.

Note: When you are using the ping utility, ensure that the pinged server has ICMP ECHO enabled, or your ping will not succeed.

Firewall Checklist

The following firewall requirements have been met:

- UDP 161 (SNMP)
- UDP 162 (SNMP trap)
- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

Load Balancing Traffic on a NetScaler

The load balancing feature distributes client requests across multiple servers to optimize resource utilization. In a real-world scenario with a limited number of servers providing service to a large number of clients, a server can become overloaded and degrade the performance of the server farm. A Citrix® NetScaler® appliance uses load balancing criteria to prevent bottlenecks by forwarding each client request to the server best suited to handle the request when it arrives.

To configure load balancing, you define a virtual server (vserver) to proxy multiple servers in a server farm and balance the load among them.

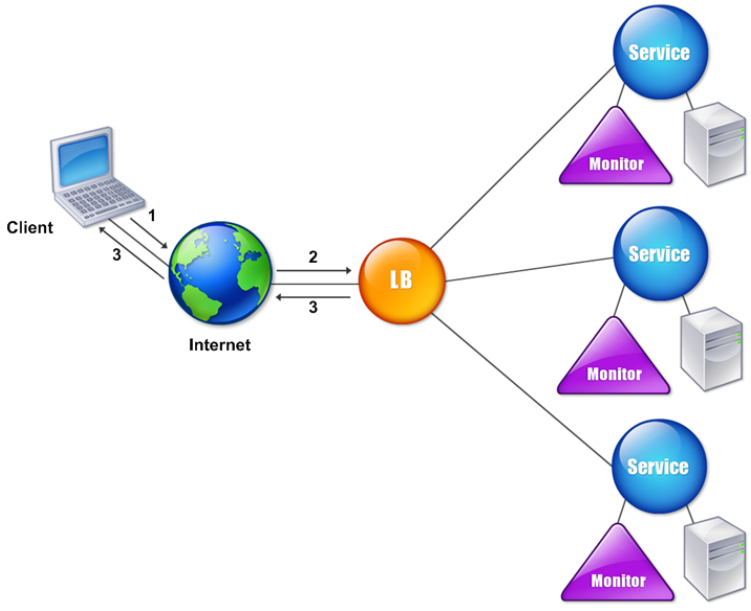
How Load Balancing Works

When a client initiates a connection to the server, a vserver terminates the client connection and initiates a new connection with the selected server, or reuses an existing connection with the server, to perform load balancing. The load balancing feature provides traffic management from Layer 4 (TCP and UDP) through Layer 7 (FTP, HTTP, and HTTPS).

The NetScaler uses a number of algorithms, called load balancing methods, to determine how to distribute the load among the servers. The default load balancing method is the Least Connections method.

A typical load balancing deployment consists of the entities described in the following figure.

Figure 1. Load Balancing Architecture



The entities function as follows:

- **Vserver**. An entity that is represented by an IP address, a port, and a protocol. The vserver IP address (VIP) is usually a public IP address. The client sends connection requests to this IP address. The vserver represents a bank of servers.

- **Service**. A logical representation of a server or an application running on a server. Identifies the server's IP address, a port, and a protocol. The services are bound to the vservers.

- **Server object**. An entity that is represented by an IP address. The server object is created when you create a service. The IP address of the service is taken as the name of the server object. You can also create a server object and then create services by using the server object.

- **Monitor**. An entity that tracks the health of the services. The NetScaler periodically probes the servers using the monitor bound to each service. If a server does not respond within a specified response timeout, and the specified number of probes fails, the service is marked DOWN. The NetScaler then performs load balancing among the remaining services.

Configuring Load Balancing

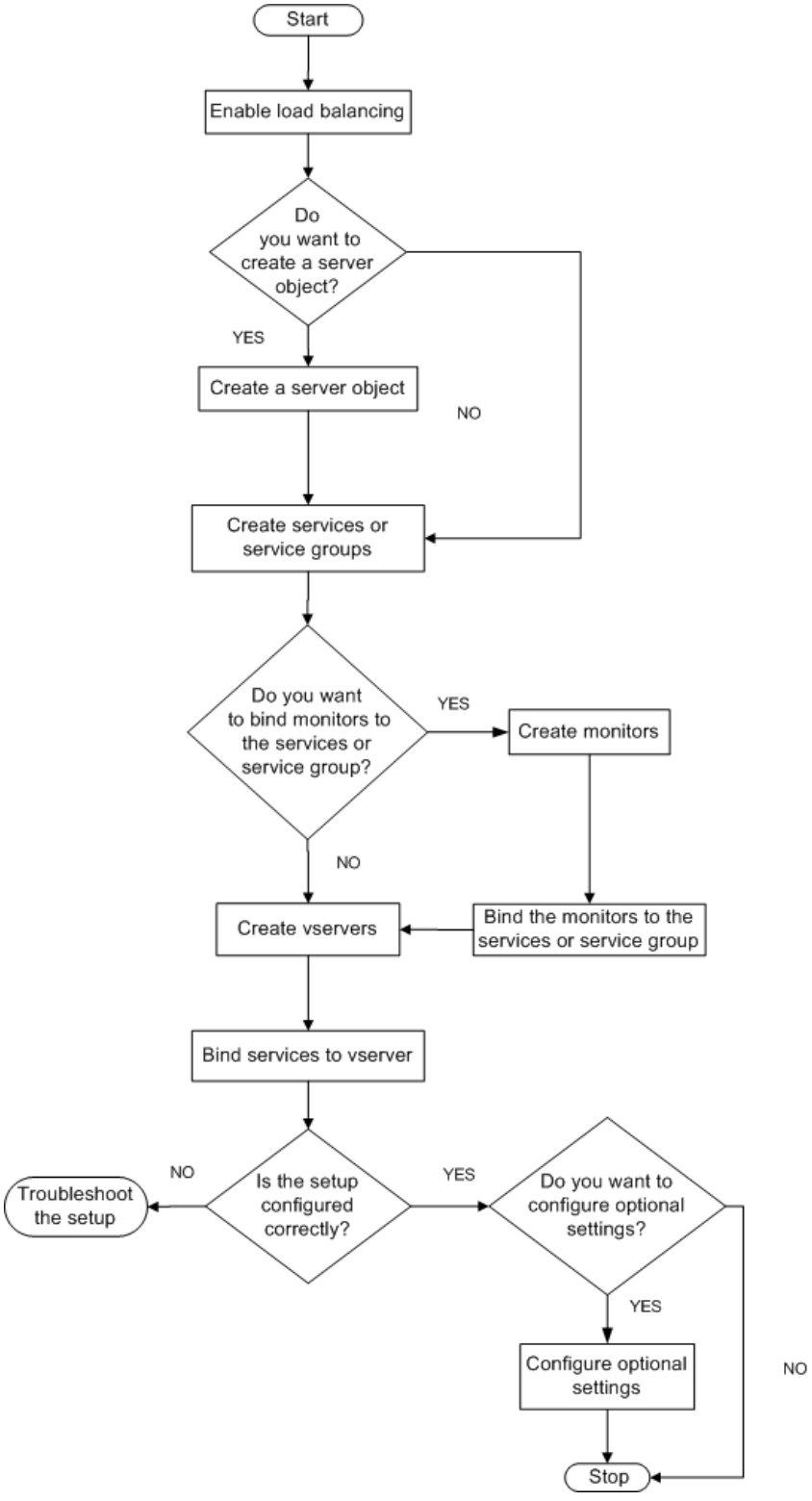
To configure load balancing, you must first create services. Then, you create vservers and bind the services to the vservers. By default, the NetScaler binds a monitor to each service. After binding the services, verify your configuration by making sure that all of the settings are correct.

Note: After you deploy the configuration, you can display statistics that show how the entities in the configuration are performing. Use the statistical utility or the `stat lb vserver <vserverName>` command.

Optionally, you can assign weights to a service. The load balancing method then uses the assigned weight to select a service. For getting started, however, you can limit optional tasks to configuring some basic persistence settings, for sessions that must maintain a connection to a particular server, and some basic configuration-protection settings.

The following flow chart illustrates the sequence of the configuration tasks.

Figure 1. Sequence of Tasks to Configure Load Balancing


Start
Enable load balancing
Do you want to create a server object?
YES
Create a server object
NO
Create services or service groups
Do you want to bind monitors to the services or service group?
YES
Create monitors
NO
Create vservers
Bind the monitors to the services or service group
Bind services to vserver
Is the setup configured correctly?
NO
Troubleshoot the setup
YES
Do you want to configure optional settings?
YES
Configure optional settings
NO
Stop

Enabling Load Balancing

Before configuring load balancing, make sure that the load balancing feature is enabled.

To enable load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable load balancing and verify that it is enabled:

- `enable feature lb`
- `show feature`

Example

```
> enable feature lb
 Done
> show feature

      Feature                Acronym          Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               OFF
3)    Load Balancing           LB               ON
.
.
.
9)    SSL Offloading           SSL              ON
.
.
.
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Load Balancing** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message, click **Yes**.

Configuring Services and a Vserver

When you have identified the services you want to load balance, you can implement your initial load balancing configuration by creating the service objects, creating a load balancing vserver, and binding the service objects to the vserver.

To implement the initial load balancing configuration by using the NetScaler command line

At the NetScaler command prompt, type the following commands to implement and verify the initial configuration:

- `add service <name> <IPaddress> <serviceType> <port>`
- `add lb vserver <vServerName> <serviceType> [<IPaddress> <port>]`
- `bind lb vserver <name> <serviceName>`
- `show service bindings <serviceName>`

Example

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
 Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
 Done
> bind lb vserver vserver-LB-1 service-HTTP-1
 Done
> show service bindings service-HTTP-1
        service-HTTP-1 (10.102.29.5:80) - State : DOWN

                1)      vserver-LB-1 (10.102.29.60:80) - State : DOWN
 Done
```

To implement the initial load balancing configuration by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the details pane, under **Getting Started**, click **Load Balancing wizard**, and follow the instructions to create a basic load balancing setup.
3. Return to the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
4. Select the vserver that you configured and verify that the parameters displayed at the bottom of the page are correctly configured.
5. Click **Open**.
6. Verify that each service is bound to the vserver by confirming that the **Active** check box is selected for each service on the **Services** tab.

Choosing and Configuring Persistence Settings

You must configure persistence on a vserver if you want to maintain the states of connections on the servers represented by that vserver (for example, connections used in e-commerce). The NetScaler then uses the configured load balancing method for the initial selection of a server, but forwards to that same server all subsequent requests from the same client.

If persistence is configured, it overrides the load balancing methods once the server has been selected. If the configured persistence applies to a service that is down, the NetScaler uses the load balancing methods to select a new service, and the new service becomes persistent for subsequent requests from the client. If the selected service is in an Out Of Service state, it continues to serve the outstanding requests but does not accept new requests or connections. After the shutdown period elapses, the existing connections are closed. The following table lists the types of persistence that you can configure.

Table 1. Limitations on Number of Simultaneous Persistent Connections

Persistence Type	Persistent Connections
Source IP, SSL Session ID, Rule, DESTIP, SRCIPDESTIP	250K
CookieInsert, URL passive, Custom Server ID	Memory limit. In case of CookieInsert, if time out is not 0, any number of connections is allowed until limited by memory.

If the configured persistence cannot be maintained because of a lack of resources on a NetScaler, the load balancing methods are used for server selection. Persistence is maintained for a configured period of time, depending on the persistence type. Some persistence types are specific to certain vservers. The following table shows the relationship.

Table 2. Persistence Types Available for Each Type of Vserver

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL Passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO

SRCIPDESTIP	N/A	N/A	YES	YES	N/A
DESTIP	N/A	N/A	YES	YES	N/A

You can also specify persistence for a group of vservers. When you enable persistence on the group, the client requests are directed to the same selected server regardless of which vserver in the group receives the client request. When the configured time for persistence elapses, any vserver in the group can be selected for incoming client requests.

Two commonly used persistence types are persistence based on cookies and persistence based on server IDs in URLs. For more information about all persistence types, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Configuring Persistence Based on Cookies

When you enable persistence based on cookies, the NetScaler adds an HTTP cookie into the Set-Cookie header field of the HTTP response. The cookie contains information about the service to which the HTTP requests must be sent. The client stores the cookie and includes it in all subsequent requests, and the NetScaler uses it to select the service for those requests. You can use this type of persistence on vservers of type HTTP or HTTPS.

The NetScaler inserts the cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

where:

- <NSC_XXXX> is the vserver ID that is derived from the vserver name.
- <ServiceIP> is the hexadecimal value of the IP address of the service.
- <ServicePort> is the hexadecimal value of the port of the service.

The NetScaler encrypts ServiceIP and ServicePort when it inserts a cookie, and decrypts them when it receives a cookie.

Note: If the client is not allowed to store the HTTP cookie, the subsequent requests do not have the HTTP cookie, and persistence is not honored.

By default, the NetScaler sends HTTP cookie version 0, in compliance with the Netscape specification. It can also send version 1, in compliance with RFC 2109.

You can configure a timeout value for persistence that is based on HTTP cookies. Note the following:

- If HTTP cookie version 0 is used, the NetScaler inserts the absolute Coordinated Universal Time (GMT) of the cookie's expiration (the expires attribute of the HTTP cookie), calculated as the sum of the current GMT time on a NetScaler, and the timeout value.
- If an HTTP cookie version 1 is used, the NetScaler inserts a relative expiration time (Max-Age attribute of the HTTP cookie). In this case, the client software calculates the actual expiration time.

Note: Most client software currently installed (Microsoft Internet Explorer and Netscape browsers) understand HTTP cookie version 0; however, some HTTP proxies understand HTTP cookie version 1.

If you set the timeout value to 0, the NetScaler does not specify the expiration time, regardless of the HTTP cookie version used. The expiration time then depends on the client software, and such cookies are not valid if that software is shut down. This persistence type

does not consume any system resources. Therefore, it can accommodate an unlimited number of persistent clients.

An administrator can use the procedure in the following table to change the HTTP cookie version.

To change the HTTP cookie version by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, click **Change HTTP Parameters**.
3. In the **Configure HTTP Parameters** dialog box, under **Cookie**, select **Version 0** or **Version 1**.

Note: For information about the parameters, see the "Load Balancing" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To configure persistence based on cookies by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure persistence based on cookies and verify the configuration:

- `set lb vserver <name> -persistenceType COOKIEINSERT`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        .
        .
        .
        Persistence: COOKIEINSERT (version 0)   Persistence Timeout: 2 min
        .
        .
        .
 Done
>
```

To configure persistence based on cookies by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure persistence (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, in the **Persistence** list, select **COOKIEINSERT**.
4. In the **Time-out (min)** text box, type the time-out value (for example, 2).
5. Click **OK**.
6. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the **Details** section at the bottom of the pane.

Configuring Persistence Based on Server IDs in URLs

The NetScaler can maintain persistence based on the server IDs in the URLs. In a technique called URL passive persistence, the NetScaler extracts the server ID from the server response and embeds it in the URL query of the client request. The server ID is an IP address and port specified as a hexadecimal number. The NetScaler extracts the server ID from subsequent client requests and uses it to select the server.

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. For more information about expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: If the server ID cannot be extracted from the client requests, server selection is based on the load balancing method.

Example: Payload Expression

The expression, URLQUERY contains sid= configures the system to extract the server ID from the URL query of a client request, after matching token sid=. Thus, a request with the URL http://www.citrix.com/index.asp?&sid;=c0a864100050 is directed to the server with the IP address10.102.29.10 and port 80.

The timeout value does not affect this type of persistence, which is maintained as long as the server ID can be extracted from the client requests. This persistence type does not consume any system resources, so it can accommodate an unlimited number of persistent clients.

Note: For information about the parameters, see the “Load Balancing” chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To configure persistence based on server IDs in URLs by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure persistence based on server IDs in URLs and verify the configuration:

- `set lb vserver <name> -persistenceType URLPASSIVE`
- `show lb vserver <name>`

  Example

```
> set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        .
        .
        .
        Persistence: URLPASSIVE Persistence Timeout: 2 min
        .
        .
        .
 Done
>
```

To configure persistence based on server IDs in URLs by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure persistence (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, in the **Persistence** list, select **URLPASSIVE**.
4. In the **Time-out (min)** text box, type the time-out value (for example, 2).
5. In the **Rule** text box, enter a valid expression. Alternatively, click **Configure** next to the **Rule** text box and use the **Create Expression** dialog box to create an expression.
6. Click **OK**.
7. Verify that the vserver for which you configured persistence is correctly configured by selecting the vserver and viewing the **Details** section at the bottom of the pane.

Configuring Features to Protect the Load Balancing Configuration

You can configure URL redirection to provide notifications of vserver malfunctions, and you can configure backup vservers to take over if a primary vserver becomes unavailable.

Configuring URL Redirection

You can configure a redirect URL to communicate the status of the NetScaler in the event that a vserver of type HTTP or HTTPS is down or disabled. This URL can be a local or remote link. The NetScaler uses HTTP 302 redirect.

Redirects can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Note: If a load balancing vserver is configured with both a backup vserver and a redirect URL, the backup vserver takes precedence over the redirect URL. In this case, a redirect is used when both the primary and backup vservers are down.

To configure a vserver to redirect client requests to a URL by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a vserver to redirect client requests to a URL and verify the configuration:

- `set lb vserver <name> -redirectURL <URL>`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
        .
        .
        .
        Redirect URL: http://www.newdomain.com/mysite/maintenance
        .
        .
        .
 Done
>
```

To configure a vserver to redirect client requests to a URL by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure URL redirection (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Advanced** tab, in the **Redirect URL** text box, type the URL (for example, `http://www.newdomain.com/mysite/maintenance`), and then click **OK**.
4. Verify that the redirect URL you configured for the server appears in the **Details** section at the bottom of the pane.

Configuring Backup Vservers

If the primary vserver is down or disabled, the NetScaler can direct the connections or client requests to a backup vserver that forwards the client traffic to the services. The NetScaler can also send a notification message to the client regarding the site outage or maintenance. The backup vserver is a proxy and is transparent to the client.

You can configure a backup vserver when you create a vserver or when you change the optional parameters of an existing vserver. You can also configure a backup vserver for an existing backup vserver, thus creating cascaded backup vservers. The maximum depth of cascading backup vservers is 10. The NetScaler searches for a backup vserver that is up and accesses that vserver to deliver the content.

You can configure URL redirection on the primary for use when the primary and the backup vservers are down or have reached their thresholds for handling requests.

Note: If no backup vserver exists, an error message appears, unless the vserver is configured with a redirect URL. If both a backup vserver and a redirect URL are configured, the backup vserver takes precedence.

To configure a backup vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a backup server and verify the configuration:

- `set lb vserver <name> [-backupVserver <string>]`
- `show lb vserver <name>`

Example

```
> set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
 Done
> show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
        .
        .
        .
        Backup: vserver-LB-2
        .
        .
        .
 Done
>
```

To set up a backup vserver by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the vserver for which you want to configure the backup vserver (for example, **vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Advanced** tab, in the **Backup Virtual Server** list, select the backup vserver (for example, **vserver-LB-2**, and then click **OK**.
4. Verify that the backup vserver you configured appears in the **Details** section at the bottom of the pane.

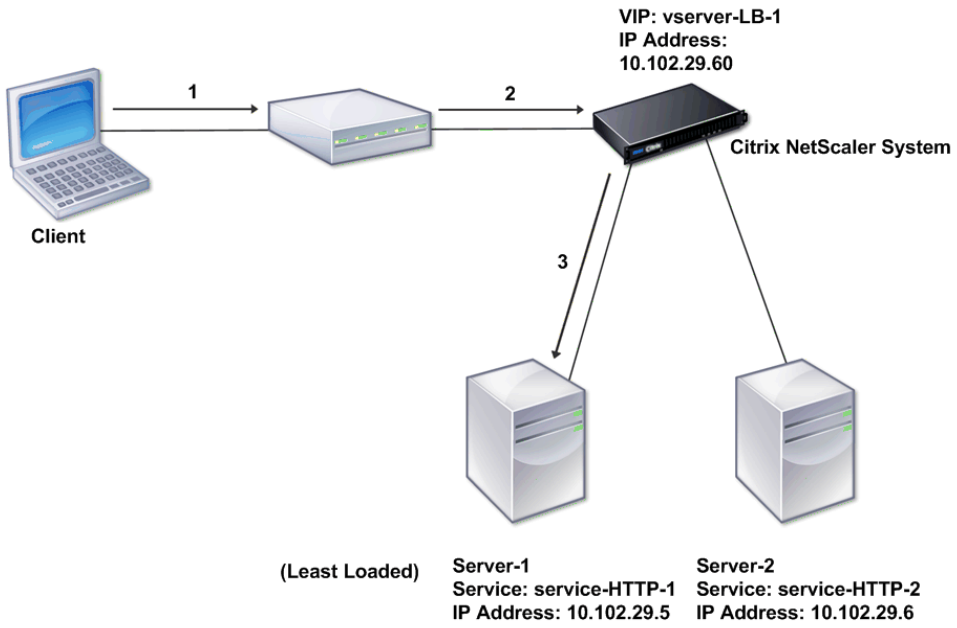
   Note: If the primary server goes down and then comes back up, and you want the backup vserver to function as the primary server until you explicitly reestablish the primary virtual server, select the **Disable Primary When Down** check box.

A Typical Load Balancing Scenario

In a load balancing setup, the NetScalers are logically located between the client and the server farm, and they manage traffic flow to the servers.

The following figure shows the topology of a basic load balancing configuration.

Figure 1. Basic Load Balancing Topology



The vserver selects the service and assigns it to serve client requests. Consider the scenario in the preceding figure, where the services service-HTTP-1 and service-HTTP-2 are created and bound to the vserver named vserver-LB-1. Vserver-LB-1 forwards the client request to either service-HTTP-1 or service-HTTP-2. The system selects the service for each request by using the Least Connections load balancing method. The following table lists the names and values of the basic entities that must be configured on the system.

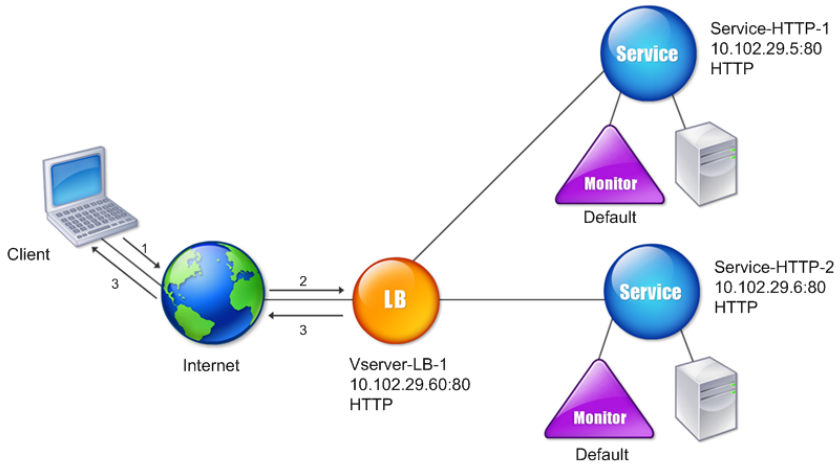
Table 1. LB Configuration Parameter Values

Entity Type	Required parameters and sample values			
	Name	IP Address	Port	Protocol
Vserver	vserver-LB-1	10.102.29.60	80	HTTP
Services	service-HTTP-1	10.102.29.5	8083	HTTP
	service-HTTP-2	10.102.29.6	80	HTTP

Monitors	Default	None	None	None

The following figure shows the load balancing sample values and required parameters that are described in the preceding table.

Figure 2. Load Balancing Entity Model



The following tables list the commands used to configure this load balancing setup by using the NetScaler command line.

Table 2. Initial Configuration Tasks

Task	Command
To enable load balancing	enable feature lb
To create a service named service-HTTP-1	add service service-HTTP-1 10.102.29.5 HTTP 80
To create a service named service-HTTP-2	add service service-HTTP-2 10.102.29.6 HTTP 80
To create a vserver named vserver-LB-1	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
To bind a service named service-HTTP-1 to a vserver named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-1
To bind a service named service-HTTP-2 to a vserver named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-2

For more information about the initial configuration tasks, see Enabling Load Balancing and Configuring Services and a Vserver.

Table 3. Verification Tasks

Task	Command
To view the properties of a vserver named vserver-LB-1	show lb vserver vserver-LB-1
To view the statistics of a vserver named vserver-LB-1	stat lb vserver vserver-LB-1
To view the properties of a service named service-HTTP-1	show service service-HTTP-1
To view the statistics of a service named service-HTTP-1	stat service service-HTTP-1
To view the bindings of a service named service-HTTP-1	show service bindings service-HTTP-1

Table 4. Customization Tasks

Task	Command
To configure persistence on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
To configure COOKIEINSERT persistence on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
To configure URLPassive persistence on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
To configure a vserver to redirect the client request to a URL on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
To set a backup vserver on a vserver named vserver-LB-1	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

For more information about configuring persistence, see Choosing and Configuring Persistence Settings. For information about configuring a vserver to redirect a client request to a URL and setting up a backup vserver, see Configuring Features to Protect the Load Balancing Configuration.

Accelerating Load Balanced Traffic by Using Compression

Compression is a popular means of optimizing bandwidth usage, and most web browsers support compressed data. If you enable the compression feature, the Citrix® NetScaler® intercepts requests from clients and determines whether the client can accept compressed content. After receiving the HTTP response from the server, the NetScaler examines the content to determine whether it is compressible. If the content is compressible, the NetScaler compresses it, modifies the response header to indicate the type of compression performed, and forwards the compressed content to the client.

NetScaler compression is a policy-based feature. A policy filters requests and responses to identify responses to be compressed, and specifies the type of compression to apply to each response. The NetScaler provides several built-in policies to compress common MIME types such as text/html, text/plain, text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint. You can also create custom policies. The NetScaler does not compress compressed MIME types such as application/octet-stream, binary, bytes, and compressed image formats such as GIF and JPEG.

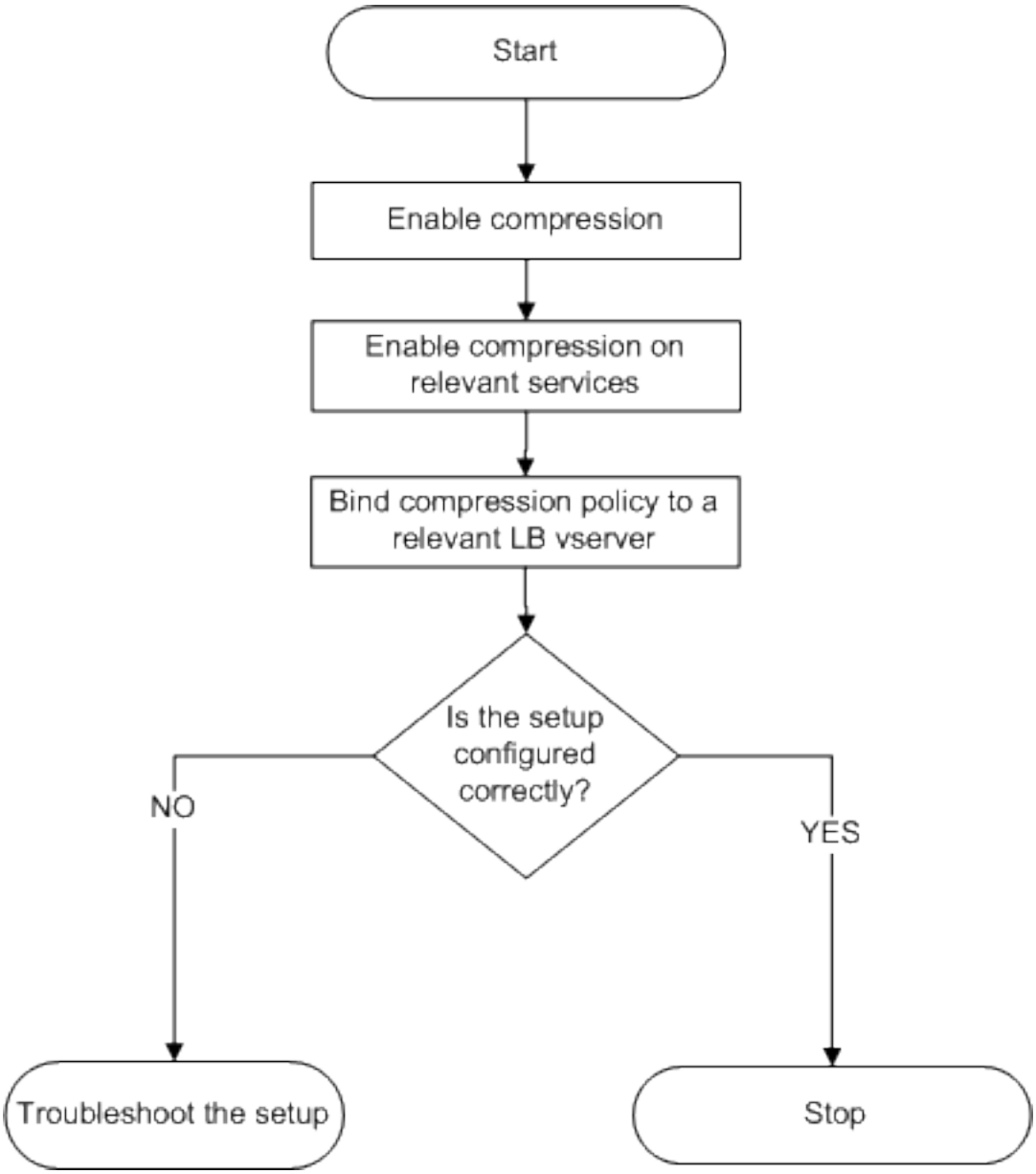
To configure compression, you must enable it globally and on each service that will provide responses that you want compressed. If you have configured vservers for load balancing or content switching, you should bind the polices to the vservers. Otherwise, the policies apply to all traffic that passes through the NetScaler.

Note: For more information about compression, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Compression Configuration Task Sequence

The following flow chart shows the sequence of tasks for configuring basic compression in a load balancing setup.

Figure 1. Sequence of Tasks to Configure Compression



Note: The steps in the above figure assume that load balancing has already been configured. For information about configuring load balancing, or for more information about services, see the *Citrix NetScaler Traffic Management Guide* at

http://support.citrix.com/article/CTX128670.

If you want to configure something other than a basic compression setup, (for example, if you need to configure optional parameters in addition to the required parameters) see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Enabling Compression

By default, compression is not enabled. You must enable the compression feature to allow compression of HTTP responses that are sent to the client.

To enable compression by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable compression and verify the configuration:

- `enable ns feature CMP`
- `show ns feature`

Example

```
> enable ns feature CMP
Done
> show ns feature

Feature                  Acronym          Status
-------                  -------          ------
1)    Web Logging              WL              ON
2)    Surge Protection          SP              OFF
.
7)    Compression Control      CMP             ON
8)    Priority Queuing         PQ              OFF
.
Done
```

To enable compression by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Compression** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Configuring Services to Compress Data

In addition to enabling compression globally, you must enable it on each service that will deliver files to be compressed. To create a service, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To enable compression on a service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable compression on a service and verify the configuration:

- `set service <name> -CMP YES`
- `show service <name>`

Example

```
> show service SVC_HTTP1
SVC_HTTP1 (10.102.29.18:80) - HTTP
State: UP
Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)
Time since last state change: 0 days, 03:03:37.200
Server Name: 10.102.29.18
Server ID : 0   Monitor Threshold : 0
Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
Use Source IP: NO
Client Keepalive(CKA): NO
Access Down Service: NO
TCP Buffering(TCPB): NO
HTTP Compression(CMP): YES
Idle timeout: Client: 180 sec   Server: 360 sec
Client IP: DISABLED
Cacheable: NO
SC: OFF
SP: OFF
Down state flush: ENABLED

1)      Monitor Name: tcp-default
State: DOWN     Weight: 1
Probes: 1095    Failed [Total: 1095 Current: 1095]
Last response: Failure - TCP syn sent, reset received.
Response Time: N/A
Done
```

To enable compression on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure compression (for example, **service-HTTP-1**), and then click **Open**.
3. On the **Advanced** tab, under **Settings**, select the **Compression** check box, and then click **OK.**
4. Verify that, when the service is selected, `HTTP Compression(CMP): ON` appears in the **Details** section at the bottom of the pane.

Binding a Compression Policy to a Virtual Server

If you bind a policy to a virtual server, the policy is evaluated only by the services associated with that virtual server. You can bind compression policies to a virtual server either from the **Configure Virtual Server (Load Balancing)** dialog box or from the **Compression Policy Manager** dialog box. This topic includes instructions to bind compression policies to a load balancing virtual server by using the **Configure Virtual Server (Load Balancing)** dialog box. For information about how you can bind a compression policy to a load balancing virtual server by using the **Compression Policy Manager** dialog box, see Configuring and Binding Policies with the Policy Manager. the "Configuring Advanced Policies" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To bind or unbind a compression policy to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind or unbind a compression policy to a load balancing virtual server and verify the configuration:

- `(bind|unbind) lb vserver <name> -policyName <string>`
- `show lb vserver <name>`

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> show lb vserver lbvip
lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Thu May 28 05:37:21 2009 (+685 ms)
Time since last state change: 19 days, 04:26:50.470
Effective State: UP
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Port Rewrite : DISABLED
No. of Bound Services :  1 (Total)       1 (Active)
Configured Method: LEASTCONNECTION
Current Method: Round Robin, Reason: Bound service's state changed to UP
Mode: IP
Persistence: NONE
```

```
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule:

Bound Service Groups:
1)      Group Name: Service-Group-1

1) Service-Group-1 (10.102.29.252: 80) - HTTP State: UP Weight:

1)      Policy : ns_cmp_msapp Priority:0
Done
```

To bind or unbind a compression policy to a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind or unbind a compression policy (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Policies** tab, click **Compression**.
4. Do one of the following:
   - To bind a compression policy, click **Insert Policy**, and then select the policy you want to bind to the virtual server.
   - To unbind a compression policy, click the name of the policy you want to unbind from the virtual server, and then click **Unbind Policy**.
5. Click **OK.**

Securing Load Balanced Traffic by Using SSL

The Citrix® NetScaler® SSL offload feature transparently improves the performance of web sites that conduct SSL transactions. By offloading CPU-intensive SSL encryption and decryption tasks from the local web server to the NetScaler, SSL offloading ensures secure delivery of web applications without the performance penalty incurred when the server processes the SSL data. Once the SSL traffic is decrypted, it can be processed by all standard services. The SSL protocol works seamlessly with various types of HTTP and TCP data and provides a secure channel for transactions using such data.

To configure SSL, you must first enable it. Then, you configure HTTP or TCP services and an SSL virtual server on the NetScaler, and bind the services to the vserver. You must also add a certificate-key pair and bind it to the SSL virtual server. If you use Outlook Web Access servers, you must create an action to enable SSL support and a policy to apply the action. An SSL virtual server intercepts incoming encrypted traffic and decrypts it by using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the NetScaler for appropriate processing.

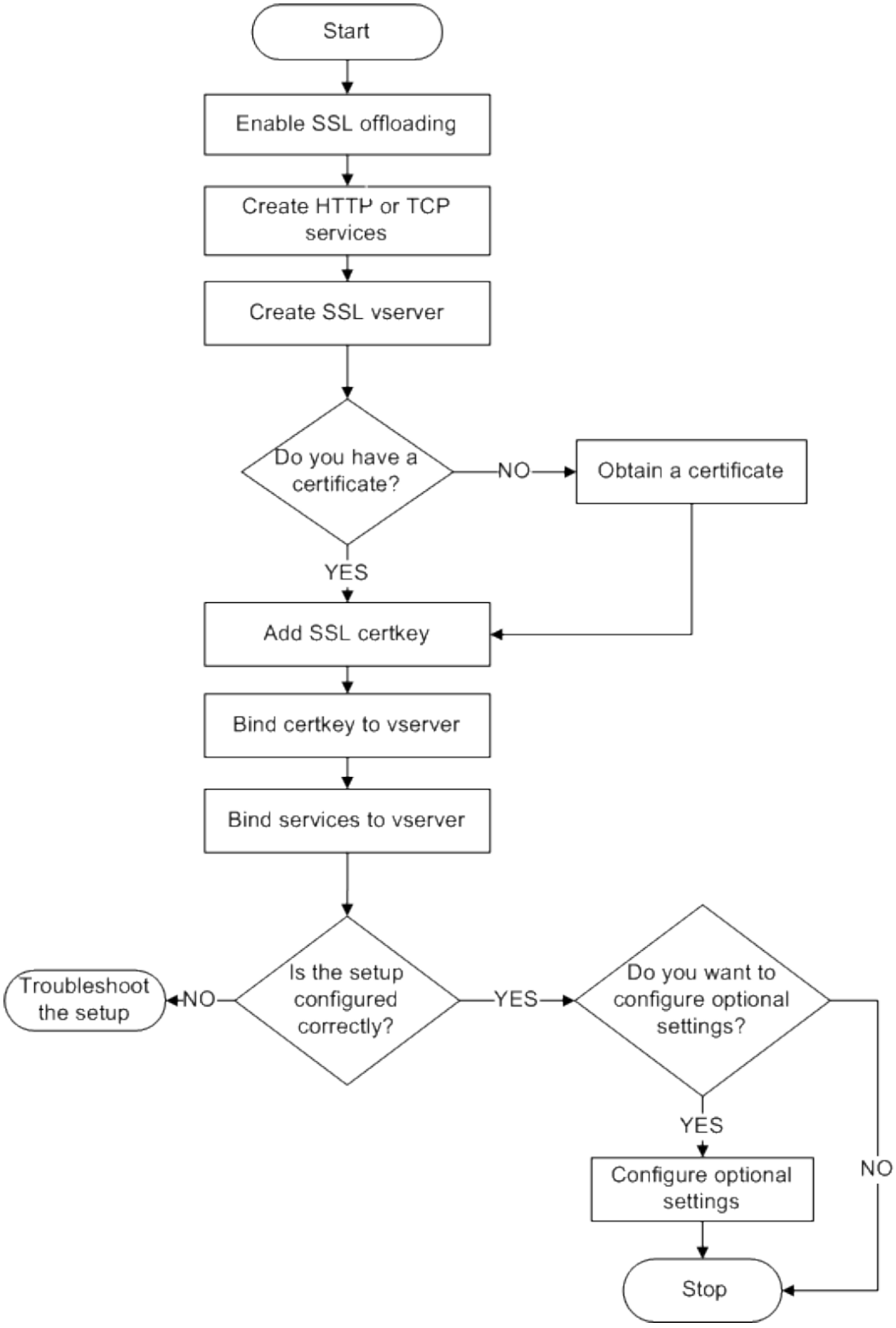
SSL Configuration Task Sequence

To configure SSL, you must first enable it. Then, you must create an SSL virtual server and HTTP or TCP services on the NetScaler. Finally, you must bind a valid SSL certificate and the configured services to the SSL virtual server.

An SSL virtual server intercepts incoming encrypted traffic and decrypts it using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the NetScaler for appropriate processing.

The following flow chart shows the sequence of tasks for configuring a basic SSL offload setup.

Figure 1. Sequence of Tasks to Configure SSL Offloading


Start
Enable SSL offloading
Create HTTP or TCP services
Create SSL vserver
Do you have a certificate?
NO
Obtain a certificate
YES
Add SSL certkey
Bind certkey to vserver
Bind services to vserver
Troubleshoot the setup
NO
Is the setup configured correctly?
YES
Do you want to configure optional settings?
YES
Configure optional settings
NO
Stop

Enabling SSL Offload

You should enable the SSL feature before configuring SSL offload. You can configure SSL-based entities on the system without enabling the SSL feature, but they will not work until you enable SSL.

To enable SSL by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable SSL Offload and and verify the configuration:

- `enable ns feature SSL`
- `show ns feature`

Example

```
> enable ns feature ssl
Done
> show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL ON
2) SurgeProtection SP OFF
3) Load Balancing LB ON . . .
 9) SSL Offloading SSL ON
10) Global Server Load Balancing GSLB ON . .
Done >
```

To enable SSL by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. Select the **SSL Offloading** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Creating HTTP Services

A service on the NetScaler represents an application on a server. Once configured, services are in the disabled state until the NetScaler can reach the server on the network and monitor its status. This topic covers the steps to create an HTTP service.

Note: For TCP traffic, perform the procedures in this and the following topics, but create TCP services instead of HTTP services.

To add an HTTP service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a HTTP service and verify the configuration:

- `add service <name> (<IP> | <serverName>) <serviceType> <port>`
- `show service <name>`

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80
 Done
> show service SVC_HTTP1
        SVC_HTTP1 (10.102.29.18:80) - HTTP
        State: UP
        Last state change was at Wed Jul 15 06:13:05 2009
        Time since last state change: 0 days, 00:00:15.350
        Server Name: 10.102.29.18
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: tcp-default
              State: UP       Weight: 1
              Probes: 4       Failed [Total: 0 Current: 0]
              Last response: Success - TCP syn+ack received.
              Response Time: N/A
```

Done

To add an HTTP service by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Services**.
2. In details pane, click **Add**.
3. In the **Create Service** dialog box, in the **Service Name**, **Server**, and **Port** text boxes, type the name of the service, IP address, and port (for example, `SVC_HTTP1`, `10.102.29.18`, and `80`).
4. In the **Protocol** list, select the type of the service (for example, `HTTP`).
5. Click **Create**, and then click **Close**. The HTTP service you configured appears in the **Services** page.
6. Verify that the parameters you configured are correctly configured by selecting the service and viewing the **Details** section at the bottom of the pane.

For more information about services, see the “Load Balancing” chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Adding an SSL-Based Vserver

In a basic SSL offloading setup, the SSL virtual server intercepts encrypted traffic, decrypts it, and sends the clear text messages to the services that are bound to the virtual server. Offloading CPU-intensive SSL processing to the NetScaler allows the back-end servers to process a greater number of requests.

To add an SSL-based vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL-based vserver and verify the configuration:

- `add lb vserver <name> <serviceType> [<IPAddress> <port>]`
- `show lb vserver <name>`

Example

```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
Done
> show lb vserver vserver-SSL-1
vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS
State: DOWN[Certkey not bound] Last state change was at Tue Jun 16 06:33:08 2009 (
Time since last state change: 0 days, 00:03:44.120
Effective State: DOWN Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
No. of Bound Services : 0 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule: Done
```

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based vserver before you enable it.

To add an SSL-based vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (SSL Offload)** dialog box, in the **Name**, **IP Address**, and **Port** text boxes, type the name of the vserver, IP address, and port (for example, `Vserver-SSL-1`, `10.102.29.50`, and `443`).
4. In the **Protocol** list, select the type of the vserver, for example, **SSL.**
5. Click **Create**, and then click **Close**.
6. Verify that the parameters you configured are correctly configured by selecting the vserver and viewing the **Details** section at the bottom of the pane. The vserver is marked as DOWN because a certificate-key pair and services have not been bound to it.

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based vserver before you enable it.

Binding Services to the SSL Vserver

After decrypting the incoming data, the SSL vserver forwards the data to the services that you have bound to the vserver.

Data transfer between the NetScaler and the servers can be encrypted or in clear text. If the data transfer between the NetScaler and the servers is encrypted, the entire transaction is secure from end to end. For more information about configuring the system for end-to-end security, see the "Secure Socket Layer (SSL) Acceleration" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To bind a service to a vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind service to the SSL vserver and verify the configuration:

- `bind lb vserver <name> <serviceName>`
- `show lb vserver <name>`

Example

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1
Done
> show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) - SSL Type:
ADDRESS State: DOWN[Certkey not bound]
Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)
Time since last state change: 0 days, 00:31:53.70
Effective State: DOWN Client Idle
Timeout: 180 sec
Down state flush: ENABLED Disable Primary Vserver On Down :
DISABLED No. of Bound Services : 1 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver IP and
Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Ru

1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
State: DOWN Weight: 1
Done
```

To bind a service to a vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, select a vserver, and then click **Open**.
3. On the **Services** tab, in the **Active** column, select the check boxes next to the services that you want to bind to the selected vserver.
4. Click **OK.**
5. Verify that the **Number of Bound Services** counter in the **Details** section at the bottom of the pane is incremented by the number of services that you bound to the vserver.

Adding a Certificate Key Pair

An SSL certificate is an integral element of the SSL Key-Exchange and encryption/decryption process. The certificate is used during SSL handshake to establish the identity of the SSL server. You can use a valid, existing SSL certificate that you have on the NetScaler, or you can create your own SSL certificate. The NetScaler supports RSA/DSA certificates of up to 4096 bits.

Note: Citrix recommends that you use a valid SSL certificate that has been issued by a trusted certificate authority. Invalid certificates and self-created certificates are not compatible with all SSL clients.

Before a certificate can be used for SSL processing, you must pair it with its corresponding key. The certificate key pair is then bound to the vserver and used for SSL processing.

To add a certificate key pair by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a certificate key pair and verify the configuration:

- `add ssl certKey <certkeyName> -cert <string> [-key <string>]`
- `show sslcertkey <name>`

Example

```
> add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
 Done
> show sslcertkey CertKey-SSL-1
      Name: CertKey-SSL-1 Status: Valid,
      Days to expiration:4811 Version: 3
      Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer: C=US,ST=California,L=San
      Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
      Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17 21:26:47 2022 GMT
      Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,CN=d efault Public Key Algori
      size: 1024
 Done
```

To add a certificate key pair by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. In the details pane, click **Add**.
3. In the **Install Certificate** dialog box, in the **Certificate-Key Pair Name** text box, type a name for the certificate key pair you want to add, for example, `Certkey-SSL-1`.
4. Under **Details**, in **Certificate File Name**, click **Browse (Appliance)** to locate the certificate. Both the certificate and the key are stored in the /nsconfig/ssl/ folder on the appliance. To use a certificate present on the local system, select **Local**.
5. Select the certificate you want to use, and then click **Select**.
6. In **Private Key File Name**, click **Browse (Appliance)** to locate the private key file. To use a private key present on the local system, select **Local**.
7. Select the key you want to use and click **Select.** To encrypt the key used in the certificate key pair, type the password to be used for encryption in the **Password** text box.
8. Click **Install**.
9. Double-click the certificate key pair and, in the **Certificate Details** window, verify that the parameters have been configured correctly and saved.

Binding an SSL Certificate Key Pair to the Vserver

After you have paired an SSL certificate with its corresponding key, you must bind the certificate key pair to the SSL vserver so that it can be used for SSL processing. Secure sessions require establishing a connection between the client computer and an SSL-based virtual server on the NetScaler. SSL processing is then carried out on the incoming traffic at the virtual server. Therefore, before enabling the SSL virtual server on the NetScaler, you need to bind a valid SSL certificate to the SSL virtual server.

To bind an SSL certificate key pair to a vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an SSL certificate key pair to a vserver and verify the configuration:

- `bind ssl vserver <vServerName> -certkeyName <string>`
- show ssl vserver <name>

  Example

```
 > bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1
Done
> show ssl vserver Vserver-SSL-1

    Advanced SSL configuration for VServer Vserver-SSL-1:
    DH: DISABLED
    Ephemeral RSA: ENABLED Refresh Count: 0
    Session Reuse: ENABLED Timeout: 120 seconds
    Cipher Redirect: ENABLED
    SSLv2 Redirect: ENABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: CertKey-SSL-1 Server Certificate
1) Cipher Name: DEFAULT
   Description: Predefined Cipher Alias
Done
```

To bind an SSL certificate key pair to a vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the vserver to which you want to bind the certificate key pair, for example, **Vserver-SSL-1**, and click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, on the **SSL Settings** tab, under **Available**, select the certificate key pair that you want to bind to the vserver (for example, **Certkey-SSL-1**), and then click **Add**.
4. Click **OK.**
5. Verify that the certificate key pair that you selected appears in the **Configured** area.

Configuring Support for Outlook Web Access

If you use Outlook Web Access (OWA) servers on your NetScaler, you must configure the NetScaler to insert a special header field, FRONT-END-HTTPS: ON, in HTTP requests directed to the OWA servers, so that the servers generate URL links as https:// instead of http://.

Note: You can enable OWA support for HTTP-based SSL vservers and services only. You cannot apply it for TCP-based SSL vservers and services.

To configure OWA support, do the following:

- Create an SSL action to enable OWA support.
- Create an SSL policy.
- Bind the policy to the SSL vserver.

Creating an SSL Action to Enable OWA Support

Before you can enable OWA support, you must creat an SSL action. SSL actions are bound to SSL policies and triggered when incoming data matches the rule specified by the policy.

To create an SSL action to enable OWA support by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

- `add ssl action <name> -OWASupport ENABLED`
- `show SSL action <name>`

```
 > add ssl action Action-SSL-OWA -OWASupport enabled
Done
> show SSL action Action-SSL-OWA
  Name: Action-SSL-OWA
  Data Insertion Action: OWA
  Support: ENABLED
Done
```

To create an SSL action to enable OWA support by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. In the details pane, on the **Actions** tab, click **Add**.
3. In the **Create SSL Action** dialog box, in the **Name** text box, type `Action-SSL-OWA`.
4. Under **Outlook Web Access**, select **Enabled**.
5. Click **Create**, and then click **Close**.
6. Verify that **Action-SSL-OWA** appears in the **SSL Actions** page.

Creating SSL Policies

SSL policies are created by using the policy infrastructure. Each SSL policy has an SSL action bound to it, and the action is carried out when incoming traffic matches the rule that has been configured in the policy.

To create an SSL policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an SSL policy and verify the configuration:

- `add ssl policy <name> -rule <expression> -reqAction <string>`
- `show ssl policy <name>`

Example

```
> add ssl policy Policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA
Done
> show ssl policy Policy-SSL-1
Name: Policy-SSL-1      Rule: ns_true
Action: Action-SSL-OWA  Hits: 0
Policy is bound to following entities
1)      PRIORITY : 0
Done
```

To create an SSL policy by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create SSL Policy** dialog box, in the **Name** text box, type the name of the SSL Policy (for example, `Policy-SSL-1`).
4. In **Request Action**, select the configured SSL action that you want to associate with this policy (for example, **Action-SSL-OWA**). The `ns_true` general expression applies the policy to all successful SSL handshake traffic. However, if you need to filter specific responses, you can create policies with a higher level of detail. For more information about configuring granular policy expressions, see Understanding Policies and Expressions.
5. In **Named Expressions**, choose the built-in general expression `ns_true` and click **Add Expression**. The expression `ns_true` now appears in the Expression text box.
6. Click **Create**, and then click **Close**.
7. Verify that the policy is correctly configured by selecting the policy and viewing the **Details** section at the bottom of the pane.

Binding the SSL Policy to an SSL Vserver

After you configure an SSL policy for OWA, bind the policy to a vserver that will intercept incoming Outlook traffic. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

To bind an SSL policy to an SSL vserver by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an SSL policy to an SSL vserver and verify the configuration:

- `bind ssl vserver <vServerName> -policyName <string>`
- `show ssl vserver <name>`

Example

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1
 Done
> show ssl vserver Vserver-SSL-1
        Advanced SSL configuration for VServer Vserver-SSL-1:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: ENABLED
        SSLv2 Redirect: ENABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1)      CertKey Name: CertKey-SSL-1 Server Certificate

1)      Policy Name: Policy-SSL-1
                        Priority: 0
1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
>
```

To bind an SSL policy to an SSL vserver by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, select the vserver (for example, **Vserver-SSL-1**), and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, click **Insert Policy**, and then select the policy that you want to bind to the SSL vserver. Optionally, you can double-click the **Priority** field and type a new priority level.
4. Click **OK.**

Features at a Glance

Citrix® NetScaler® features can be configured independently or in combinations to address specific needs. Although some features fit more than one category, the numerous NetScaler features can generally be categorized as application switching and traffic management features, application acceleration features, application security and firewall features, and an application visibility feature.

To understand the order in which the features perform their processing, see Processing Order of Features.

Application Switching and Traffic Management Features

SSL Offloading

Transparently offloads SSL encryption and decryption from web servers, freeing server resources to service content requests. SSL places a heavy burden on an application's performance and can render many optimization measures ineffective. SSL offload and acceleration allow all the benefits of Citrix Request Switching technology to be applied to SSL traffic, ensuring secure delivery of web applications without degrading end-user performance.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Access Control Lists

Compares incoming packets to the Access Control Lists (ACLs). If a packet matches an ACL rule, the action specified in the rule is applied to the packet. Otherwise, the default action (ALLOW) is applied and the packet is processed normally by the system. For the system to compare incoming packets to the ACLs, you need to apply the ACLs. All ACLs are enabled by default, but you have to apply them in order for the NetScaler to compare incoming packets against them. If an ACL is not required to be a part of the lookup table, but still needs to be retained in the configuration, it should be disabled before the ACLs are applied. A NetScaler does not compare incoming packets to disabled ACLs.

For more information, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

Load Balancing

Load balancing decisions are based on a variety of algorithms, including round robin, least connections, weighted least bandwidth, weighted least packets, minimum response time, and hashing based on URL, domain source IP or destination IP. Both the TCP and UDP protocols are supported, so the NetScaler can load balance all traffic that uses those protocols as the underlying carrier (for example, HTTP, HTTPS, UDP, DNS, FTP, NNTP, and general firewall traffic). In addition, the NetScaler can maintain session persistence based on source IP, cookie, server, group or SSL session. It allows users to apply custom Extended Content Verification (ECV) to servers, caches, firewalls and other infrastructure devices to ensure that these systems are functioning properly and are providing the right content to users. It can also perform health checks using ping, TCP, or HTTP URL, and the user can create monitors based on Perl scripts.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Content Switching

Determines which server to send the request to based on the configured content switching policies. Policy rules can be configured based on the IP address, URL, and HTTP headers. This allows switching decisions to be based on user and device characteristics such as who the user is, what type of agent is being used, and what content the user requested.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Global Server Load Balancing (GSLB)

Extends the traffic management capabilities of a NetScaler to include distributed Internet sites and global enterprises. Whether installations are spread across multiple network locations or multiple clusters in a single location, the NetScaler maintains availability and distributes traffic across them. It makes intelligent DNS decisions to prevent users from being sent to a site that is down or overloaded. When the proximity-based GSLB method is enabled, the NetScaler can make load balancing decisions based on the proximity of the client's local DNS server (LDNS) in relation to different sites. The main benefit of the proximity-based GSLB method is faster response time resulting from the selection of the closest available site.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Dynamic Routing

Enables routers to obtain topology information, routes, and IP addresses from neighboring routers automatically. When dynamic routing is enabled, the corresponding routing process listens to route updates and advertises routes. The routing processes can also be placed in passive mode. Routing protocols enable an upstream router to load balance traffic to identical vservers hosted on two standalone NetScaler units using the Equal Cost Multipath technique.

For more information, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

Link Load Balancing

Load balances multiple WAN links and provides link failover, further optimizing network performance and ensuring business continuity. Ensures that network connections remain highly available, by applying intelligent traffic control and health checks to distribute traffic efficiently across upstream routers. Identifies the best WAN link to route both incoming and outbound traffic based on policies and network conditions, and protects applications against WAN or Internet link failure by providing rapid fault detection and failover.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

TCP Optimization

Transfers certain TCP tasks from your managed servers to the NetScaler, reducing CPU load on your managed servers and improving performance.

For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Web Interface on NetScaler

Provides access to XenApp and XenDesktop resources, which include applications, content, and desktops. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in. The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance.

Note: Web Interface is supported only on NetScaler nCore releases.

For more information, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Load Balancing of Branch Repeaters for WAN Optimization

To provide high-scale WAN optimization, the branch repeaters deployed at data centers can be load balanced through NetScaler appliances. The bandwidth and number of concurrent sessions can be improved significantly.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

OpenCloud Bridge

The Citrix NetScaler® OpenCloud Bridge™ feature, a fundamental part of the Citrix® OpenCloud framework, is a tool used to build a cloud-extended data center. The OpenCloud Bridge enables you to connect one or more NetScaler appliances or VPXs on the cloud-to your network without reconfiguring your network. Cloud hosted applications appear as though they are running on one contiguous enterprise network.The primary purpose of the OpenCloud Bridge is to enable companies to move their applications to the cloud while reducing costs and the risk of application failure. In addition, the OpenCloud Bridge increases network security in cloud environments. An OpenCloud Bridge is a Layer-2 network bridge that connects a NetScaler appliance or VPX on a cloud instance to a NetScaler appliance or VPX on your LAN. The connection is made through a tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. Then Internet Protocol security (IPsec) protocol suite is used to secure the communication between the peers in the OpenCloud Bridge.

For more information, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

DataStream™

The DataStream feature of NetScaler provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a

decision based on SQL query parameters, such as user name and database name, command parameters. You can further configure monitors to track the state of database servers.

The advanced policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with `MYSQL.CLIENT` and `MYSQL.REQ`) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with `MYSQL.RES`) to evaluate server responses to user-configured health monitors. For more information about these expressions, see the "Evaluating Connections to Database Servers" chapter in the *Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: DataStream is supported only for MySQL databases. For information about the DataStream feature, see the "DataStream" chapter in the *Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Application Acceleration Features

AppCompress

Provides transparent compression for HTML and text files using the gzip compression protocol. The typical 4:1 compression ratio yields up to 50% reduction in bandwidth requirements out of the data center. This also results in significantly improved end-user response time by reducing the amount of data that must be delivered to the user's browser.

For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Cache Redirection

Manages the flow of traffic to a reverse proxy, transparent proxy, or forward proxy cache farm. Inspects all requests, and identifies non-cacheable requests and sends them directly to the origin servers over persistent connections. By intelligently redirecting non-cacheable requests back to the origin web servers, the NetScaler frees cache resources and increases cache hit rates while reducing overall bandwidth consumption and response delays for these requests.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

AppCache

Helps optimize web content and application data delivery by providing a fast in-memory HTTP/1.1 and HTTP/1.0 compliant web caching for both static and dynamic content. This on-board cache stores the results of incoming application requests even when an incoming request is secured or the data compressed, and then reuses the data to fulfill subsequent requests for the same information. By serving data directly from the on-board cache, the NetScaler can reduce page regeneration times by eliminating the need to funnel static and dynamic content requests to the server.

For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

TCP Buffering

Buffers the server's response and delivers it to the client at the client's speed, thus offloading the server faster and thereby improving the performance of web sites. For more information, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Application Security and Firewall Features

Denial of Service Attack (DoS) Defense

Detects and stops malicious distributed denial-of-service (DDoS) attacks and other types of malicious attacks before they reach your servers, preventing them from affecting network and application performance. The NetScaler identifies legitimate clients and elevates their priority, leaving suspect clients unable to consume a disproportionate percentage of resources and cripple your site. A NetScaler provides application-level protection from the following types of malicious attacks:

- SYN flood attacks
- Pipeline attacks
- Teardop attacks
- Land attacks
- Fraggle attacks
- Zombie connection attacks

The NetScaler aggressively defends against these types of attacks by preventing the allocation of server resources for these connections. This insulates servers from the overwhelming flood of packets associated with these events.

The NetScaler also protects network resources from ICMP based attacks by using ICMP rate limiting and aggressive ICMP packet inspection. It performs strong IP reassembly, drops a variety of suspicious and malformed packets, and applies Access Control Lists (ACLs) to site traffic for further protection.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Content Filtering

Provides protection from malicious attacks for web sites at the Layer 7 level. The NetScaler inspects each incoming request according to user-configured rules based on HTTP headers, and performs the action the user configured. Actions can include resetting the connection, dropping the request, or sending an error message to the user's browser. This allows the NetScaler to screen unwanted requests and reduces your servers' exposure to attacks.

This feature can also analyze HTTP GET and POST requests and filter out known bad signatures, allowing it to defend your servers against HTTP-based attacks such as variants of the Nimda and Code Red viruses.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Responder

Functions like an advanced filter and can be used to generate responses from the NetScaler to the client. Some common uses of this feature are generation of redirect responses, user defined responses or resets.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Rewrite

Modifies HTTP headers and body text. You can use it to add HTTP headers to an HTTP request or response, make modifications to individual HTTP headers, or delete HTTP headers. It also lets you modify the HTTP body in requests and responses.

When the NetScaler receives a request or sends a response, it checks for rewrite rules, and if applicable rules exist, it applies them to the request or response before passing it on to the web server or client computer.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Priority Queuing

Prioritizes user requests to ensure that the most important traffic is serviced first during surges in request volume. You can establish priority based on request URLs, cookies, or a variety of other factors. The NetScaler places requests in a three-tier queue based on their configured priority, enabling business-critical transactions to flow smoothly even during surges or site attacks.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Surge Protection

Regulates the flow of user requests to servers and controls the number of users that can simultaneously access the resources on the servers, queuing any additional requests once your servers have reached their capacity. By controlling the rate at which connections can be established, the NetScaler blocks surges in requests from being passed on to your servers, thus preventing site overload.

For more information, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

Access Gateway

Citrix Access Gateway is a secure application access solution that provides administrators granular application-level policy and action controls to secure access to applications and data while allowing users to work from anywhere. It gives IT administrators a single point of control and tools to help ensure compliance with regulations and the highest levels of information security across and outside the enterprise. At the same time, it empowers users with a single point of access—optimized for roles, devices, and networks—to the

enterprise applications and data they need. This unique combination of capabilities helps maximize the productivity of today's mobile workforce.

For more information, see Access Gateway.

Application Firewall

Protects applications from misuse by hackers and malware, such as cross site scripting attacks, buffer overflow attacks, SQL injection attacks, and forceful browsing, by filtering traffic between each protected web server and users that connect to any web site on that web server. The Application Firewall examines all traffic for evidence of attacks on web server security or misuse of web server resources, and takes the appropriate action to prevent these attacks from succeeding.

For more information, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Application Visibility Feature

EdgeSight for NetScaler

Support for application performance monitoring based on end user experience. This solution leverages the HTML injection feature to obtain various time values, which are used by EdgeSight server for analysis and report generation. EdgeSight for NetScaler provides a way to monitor the performance benefits of a NetScaler and determine potential bottlenecks in a network.

For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Enhanced Application Visiblity Using AppFlow

The Citrix® NetScaler® appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

Note: The AppFlow feature is supported only by Citrix® nCore™ software.

Licensing, Upgrading, and Downgrading

The following topics describe the migration instructions for setting up a new version of a NetScaler with a list of all new and deprecated commands, parameters, and SNMP OIDs.

New and Deprecated Commands, Parameters, and SNMP OIDs	Lists the foundational changes that affect the base system and its configuration, including new and deprecated commands, parameters, and SNMP OIDs.
Upgrading or Downgrading the System Software	Describes the licensing framework and the procedure for upgrading or downgrading the system software across releases and within a release.

New and Deprecated Commands, Parameters, and SNMP OIDs

Welcome to the NetScaler 9.3 system software release. There are new commands, parameters, and SNMP OIDs in this release. Some commands, parameters, and SNMP OIDs will be deprecated in this release. For complete descriptions of the new commands and parameters, see the *Citrix NetScaler Command Reference Guide* at http://support.citrix.com/article/CTX128678. For complete descriptions of the SNMP OIDs, see the *Citrix NetScaler SNMP OID Reference Guide* at http://support.citrix.com/article/CTX128676.

New and Deprecated Commands, Parameters, and SNMP OIDs

Welcome to the NetScaler 9.3 system software release. There are new commands, parameters, and SNMP OIDs in this release. Some commands, parameters, and SNMP OIDs will be deprecated in this release. For complete descriptions of the new commands and parameters, see the *Citrix NetScaler Command Reference Guide* at http://support.citrix.com/article/CTX128678. For complete descriptions of the SNMP OIDs, see the *Citrix NetScaler SNMP OID Reference Guide* at http://support.citrix.com/article/CTX128676.

New Commands

The following table lists all the new commands in release 9.3.

Table 1. New Commands

Command Group	Command
Basic	show servicegroupbindings count servicegroupbindings unset serviceGroup
AAA	unset aaa parameter
Application Firewall	update appfw customSettings update appfw xmlerrorpage update appfw htmlerrorpage
Authentication	add authentication negotiateAction rm authentication negotiateAction set authentication negotiateAction show authentication negotiateAction add authentication negotiatePolicy rm authentication negotiatePolicy set authentication negotiatePolicy show authentication negotiatePolicy
Authorization	stat authorization policybinding stat authorization policylabel
Cache Redirection	stat cache policybinding stat cache policylabel

DB	add db user rm db user set db user show db user
Domain Name Service (DNS)	stat dns records
Load Balancing	set lb parameter show lb parameter
Network	set arpparam show arpparam stat bridge add netbridge rm netbridge show netbridge bind netbridge unbind netbridge
NetScaler	show ns savedConfig set ns encryptionParams show ns encryptionParams show ns rollbackcmd stat ns memory
ACL	clear ns simpleacl6 add ns simpleacl6 rm ns simpleacl6 show ns simpleacl6 stat ns simpleacl6

AppFlow	add ns appflowCollector rm ns appflowCollector show ns appflowCollector set ns appflowParam show ns appflowParam
NTP	show ntp status
Policy	add policy stringmap rm policy stringmap set policy stringmap bind policy stringmap unbind policy stringmap show policy stringmap
Responder	stat responder policybinding stat responder policylabel
Rewrite	stat rewrite policybinding stat rewrite policylabel
SNMP	set snmp manager
System	stat system memory set system group
Web Interface	install wi package uninstall wi package add wi site rm wi site set wi site bind wi site unbind wi site show wi site

Cloud Bridge	add ipsec peer show ipsec peer rm ipsec peer set ipsec parameter unset ipsec parameter show ipsec parameter

Deprecated Commands

The following commands are deprecated in release 9.3.

- rm appfw customSettings
- show appfw customSettings
- import appfw customSettings
- update appfw customSettings

New Parameters

The following table lists all the new parameters in release 9.3.

Table 1. New Parameters

Command Group	Command
Basic	show servicegroupbindings [**-servicegroupname**] stat serviceGroupMember [**-servername**] disable server [**-graceful** (YES | NO)] add service [**-healthmonitor** (YES | NO)] [**-hashid** <positive_integer>] [**-appflowlog** (ENABLED | DISABLED)] set service [**-healthmonitor** (YES | NO)] [**-hashid** <positive_integer>] [**-appflowlog** (YES | NO)] unset service [**-hashid** <positive_integer>] [**-appflowlog** (ENABLED | DISABLED)] disable service [**-graceful** (YES | NO)] add serviceGroup [**-healthmonitor** (YES | NO)] [**-appflowlog** (ENABLED | DISABLED)] set serviceGroup [**-healthmonitor** (YES | NO)] [**-appflowlog** (ENABLED | DISABLED)] unset serviceGroup [**-servicegroupname**] [**-servername**] [**-maxclient**] [**-maxreq**] [**-cacheable**] [**-cip**] [**-usip**] [**-useproxyport**] [**-sc**] [**-sp**] [**-rtspsessionidremap**] [**-clttimeout**] [**-svrtimeout**] [**-cka**] [**-tcpb**] [**-cmp**] [**-maxbandwidth**] [**-monthreshold**] [**-appflowlog**] disable serviceGroup [**-graceful** (YES | NO)]
AAA	unset aaa parameter [**-defaultauthtype**] [**-maxaaausers**]

Application Firewall	set appfw settings [**-cookiepostencryptprefix** <string>] add appfw profile [**-cookietransforms** (ON | OFF)] [**-cookieencryption** <cookieEncryption>] [**-cookieproxying** (none | sessionOnly)] [**-addcookieflags** <addCookieFlags>] [**-striphtmlcomments** <stripHtmlComments>] [**-fileuploadmaxnum** <positive_integer>] set appfw profile [**-cookietransforms** (ON | OFF)] [**-cookieencryption** <cookieEncryption>] [**-cookieproxying** (none | sessionOnly)] [**-addcookieflags** <addCookieFlags>] [**-striphtmlcomments** <stripHtmlComments>] [**-fileuploadmaxnum** <positive_integer>] bind appfw profile [**-xmlsqlinjection** <string>] [**-xmlxss** <string>] unbind appfw profile [**-xmlsqlinjection** <string>] [**-xmlxss** <string>] import appfw wsdl [**-comment** <string>] [**-overwrite**] import appfw customSettings [**-comment** <string>] [**-overwrite**] import appfw xmlschema [**-comment** <string>] [**-overwrite**] import appfw xmlerrorpage [**-comment** <string>] [**-overwrite**] import appfw htmlerrorpage [**-comment** <string>] [**-overwrite**] update appfw customSettings [**-name**] [**-type**] update appfw xmlerrorpage [**-name**] [**-type**] update appfw htmlerrorpage [**-name**] [**-type**]

Audit	add audit syslogAction [**-appflowexport** (ENABLED | DISABLED)] set audit syslogAction [**-appflowexport** (ENABLED |DISABLED)] unset audit syslogAction [**-appflowexport** (ENABLED | DISABLED)] set audit syslogParams [**-appflowexport** (ENABLED |DISABLED)] unset audit syslogParams [**-appflowexport**] add audit nslogAction [**-appflowexport**] set audit nslogAction [**-appflowexport** (ENABLED | DISABLED)] unset audit nslogAction [**-appflowexport**] set audit nslogParams [**-appflowexport** (ENABLED |DISABLED)] unset audit nslogParams [**-appflowexport**]

Authentication	add authentication negotiateAction [**-name**] [**-domain** <string>] [**-domainuser** <string>] [**-domainuserpasswd**] [**-ou** <string>] rm authentication negotiateAction [**-name**] set authentication negotiateAction [**-name**] [**-domain** <string>] [**-domainuser** <string>] [**-domainuserpasswd**] [**-ou** <string>] show authentication negotiateAction [**-name**] add authentication negotiatePolicy [**-name**] [**-rule**] [**-reqaction**] rm authentication negotiatePolicy [**-name**] set authentication negotiatePolicy [**-name**] [**-rule** <expression>] [**-reqaction** <string>] show authentication negotiatePolicy [**-name**] add authentication vserver [**-appflowlog** (ENABLED | DISABLED)] set authentication vserver [**-appflowlog** (ENABLED | DISABLED)]
Authorization	stat authorization policybinding [**-name**] stat authorization policylabel [**-labelname**] [**-bindings**]
Cache Redirection	stat cache policybinding [**-name**] stat cache policylabel [**-labelname**] [**-bindings**] add cr vserver [**-useportrange** (ON | OFF)] [**-appflowlog** (ENABLED |DISABLED)] set cr vserver [**-useportrange** (ON | OFF)] [**-appflowlog** (ENABLED |DISABLED)] unset cr vserver [**-useportrange**] [**-appflowlog**]

Content Switching	add cs vserver [**-mysqlprotocolversion** <positive_integer>] [**-mysqlserverversion** <string>] [**-mysqlcharacterset** <positive_integer>] [**-mysqlservercapabilities** <positive_integer>] [**-appflowlog**] set cs vserver [**-mysqlprotocolversion**] [**-mysqlserverversion**] [**-mysqlcharacterset**] [**-mysqlservercapabilities**] [**-appflowlog** (ENABLED | DISABLED)] unset cs vserver [**-mysqlprotocolversion**] [**-mysqlserverversion**] [**-mysqlcharacterset**] [**-mysqlservercapabilities**] [**-appflowlog**]
DB	add db user [**-username**] [**-password**] rm db user [**-username**] set db user [**-username**] [**-password**] show db user [**-username**] [**-loggedin**]
Domain Name Service (DNS)	stat dns records [**-dnsrecordtype**] add dns policy [**-cachebypass** (YES | NO)] set dns policy [**-cachebypass** (YES | NO)]
Global Server Load Balancing	add gslb service [**-healthmonitor** (YES | NO)] [**-hashid** hashid] [**-appflowlog** (ENABLED |DISABLED)] set gslb service [**-healthmonitor** (YES | NO)] [**-hashid** <positive_integer>] [**-appflowlog** (ENABLED | DISABLED)] add gslb vserver [**-dnsrecordtype** <dnsRecordType>] [**-appflowlog** (ENABLED | DISABLED)] set gslb vserver [**-dnsrecordtype** <dnsRecordType>] [**-appflowlog** (ENABLED | DISABLED)]

Load Balancing	add lb monitor [**-validatecred** (YES | NO)] [**-domain** <string>] [**-evalrule** <expression>] [**-mssqlprotocolversion** <mssqlProtocolVersion>] set lb monitor [**-validatecred** (YES | NO)] [**-domain** <string>] [**-evalrule** <expression>] add lb vserver [**-mysqlprotocolversion** <positive_integer>] [**-mysqlserverversion** <string>] [**-mysqlcharacterset** <positive_integer>] [**-mysqlservercapabilities** <positive_integer>] [**-appflowlog** (ENABLED |DISABLED)] set lb vserver [**-mysqlprotocolversion** <positive_integer>] [**-mysqlserverversion** <string>] [**-mysqlcharacterset** <positive_integer>] [**-mysqlservercapabilities** <positive_integer>] [**-appflowlog** (ENABLED | DISABLED)] unset lb vserver [**-redirecturl**] [**-mysqlprotocolversion**] [**-mysqlserverversion**] [**-mysqlcharacterset**] [**-mysqlservercapabilities**] [**-appflowlog**] set lb sipParameters [**-sip503ratethreshold** <positive_integer>] set lb parameter [**-httponlycookieflag** (ENABLED | DISABLED)] [**-useportforhashlb** (YES | NO)] [**-preferdirectroute** (YES | NO)] [**-startuprrfactor** <positive_integer>] [**-monitorskipmaxclient** (ENABLED | DISABLED)] [**-monitorconnectionclose** (RESET | FIN)] show lb parameter [**-format**] [**-level**]

Network	set arpparam [**-timeout** <positive_integer>] add vlan [**-aliasname** <string>] set vlan [**-aliasname** <string>] set ipTunnelParam [**-srciproundrobin** (YES | NO)] add ipTunnel [**-secure** (YES | NO)] add netbridge [**-name**] rm netbridge [**-name**] show netbridge [**-name**] bind netbridge [**-name**] [**-tunnel** <string> ...] [**-vlan** <positive_integer> ...] [**-ipaddress** <ip_addr|ipv6_addr] unbind netbridge [**-name**] [**-tunnel** <string> ...] [**-vlan** <positive_integer> ...] [**-ipaddress** <ip_addr|ipv6_addr]
NetScaler	show ns savedConfig [**-format**] [**-level**] set ns config [**-crportrange** <int[-int]>] unset ns config [**-crportrange**] set ns tcpParam [**-learnvsvrmss** (ENABLED | DISABLED)] diff ns config [**-ignoredevicespecific**] add ns tcpProfile [**-mss** <positive_integer>] [**-buffersize** <positive_integer>] set ns tcpProfile [**-mss** <positive_integer>] [**-buffersize**] unset ns tcpProfile [**-buffersize** <positive_integer>] set ns encryptionParams [**-method** <method>] show ns rollbackcmd [**-filename** <input_filename>] [**-outtype** (cli | xml)] [**-outfilename**] stat ns memory [**-pool**]

ACL	add ns simpleacl6 [**-aclname**] [**-aclaction**] [**-srcipv6** <ipv6_addr|null>] [**-destport** <port>] [**-ttl** <positive_integer>] rm ns simpleacl6 [**-aclname**] show ns simpleacl6 [**-aclname**]
AppFlow	add ns appflowCollector [**-name**] [**-ipaddress** <ip_addr>] [**-port** <port>] rm ns appflowCollector [**-name**] show ns appflowCollector [**-name**] set ns appflowParam [**-templaterefresh** <secs>] [**-udppmtu** <positive_integer>] [**-httpurl** (ENABLED | DISABLED)] [**-httpcookie** (ENABLED | DISABLED)] [**-httpreferer** (ENABLED | DISABLED)] [**-httpmethod** (ENABLED | DISABLED)] [**-httphost** (ENABLED | DISABLED)] [**-httpuseragent** (ENABLED | DISABLED)] [**-clienttrafficonly** (YES | NO)]
NTP	set ntp server [**-preferredntpserver** (YES | NO)]
Policy	add policy stringmap [**-name**] [**-comment** <string>] rm policy stringmap [**-name**] set policy stringmap [**-name**] [**-comment** <string>] bind policy stringmap [**-name**] [**-key**] unbind policy stringmap [**-name**] [**-key**] show policy stringmap [**-name**]
Responder	stat responder policybinding [**-name**] add responder policy [**-logaction** <string>] set responder policy [**-logaction** <string>] unset responder policy [**-logaction**] stat responder policylabel [**-labelname**] [**-bindings**]

Rewrite	stat rewrite policybinding [**-name**] add rewrite action [**-bypasssafetycheck** (YES | NO)] stat rewrite policylabel [**-labelname**] [**-bindings**]
SNMP	add snmp manager [**-ipaddress**] [**-domainresolveretry** <integer>] rm snmp manager [**-ipaddress**] set snmp manager [**-ipaddress**] [**-netmask**] <netmask> [**-domainresolveretry** <integer>] show snmp manager [**-ipaddress**]
SSL	convert ssl pkcs12 [**-password**] [**-pempassphrase**] create ssl certReq [**-countryname** <string>] [**-statename** <string>] [**-organizationname** <string>] [**-organizationunitname** <string>] [**-localityname** <string>] [**-commonname** <string>] [**-emailaddress** <string>] [**-challengepassword**] [**-companyname** <string>] create ssl cert [**-pempassphrase**] set ssl parameter [**-pushflag** <positive_integer>] [**-dropreqwithnohostheader** (YES | NO)] [**-pushenctriggertimeout** <positive_integer>] set ssl service [**-pushenctrigger** <pushEncTrigger>] set ssl vserver [**-pushenctrigger** <pushEncTrigger>] add ssl ocspResponder [**-insertclientcert** (YES | NO)] set ssl ocspResponder [**-insertclientcert** (YES | NO)] unset ssl ocspResponder [**-insertclientcert** (YES | NO)]

System	add system user [**-promptstring** <string>] set system user [**-promptstring** <string>] add system group [**-promptstring** <string>] set system group [**-groupname**] [**-promptstring** <string>] set system parameter [**-promptstring** <string>]
Traffic Management	add tm sessionAction [**-persistentcookie** (ENABLED | DISABLED)] [**-persistentcookievalidity** <mins>] set tm sessionAction [**-persistentcookie** (ENABLED | DISABLED)] [**-persistentcookievalidity** <positive_integer>] set tm sessionParameter [**-persistentcookie** (ENABLED | DISABLED)] [**-persistentcookievalidity** <positive_integer>] unset tm sessionParameter [**-persistentcookie**] add tm trafficAction [**-persistentcookie** (ENABLED | DISABLED)] set tm trafficAction [**-persistentcookie** (ENABLED | DISABLED)]
Network	ping6 [**-v** <vlanid>]
Access Gateway	add vpn vserver [**-appflowlog** (ENABLED | DISABLED)] set vpn vserver [**-appflowlog** (ENABLED | DISABLED)]

Web Interface	install wi package [**-jre** <URL>] [**-wi** <URL>] [**-clearsites**] [**-maxsites** <maxSites>] add wi site [**-sitepath**] [**-agurl**] [**-wiauthenticationmethods** (Explicit | Anonymous)] [**-sitetype** (XenAppWeb | XenAppServices)] [**-publishedresourcetype** <publishedResourceType>] [**-kioskmode** (ON | OFF)] rm wi site [**-sitepath**] set wi site [**-sitepath**] [**-agurl** <string>] [**-staurl** <string>] [**-sessionreliability** (ON | OFF)] [**-usetwotickets** (ON | OFF)] [**-secondstaurl** <string>] [**-wiauthenticationmethods** (Explicit | Anonymous)] [**-authenticationpoint** (WebInterface | AccessGateway)] [**-publishedresourcetype** <publishedResourceType>] [**-kioskmode** (ON | OFF)] bind wi site [**-sitepath**] [**-farmname**] [**-xmlserveraddresses**] [**-xmlport** <positive_integer>] [**-transport** <transport>] [**-loadbalance** (ON | OFF)] unbind wi site [**-sitepath**] [**-farmname**] show wi site [**-sitepath**]
Cloud Bridge	add ipsec peer [**-name**] [**-fqdn**] [**-peerfqdn**] [**-localip**] [**-peerip**] [**-encalgo** (AES | 3DES)] [**-hashalgo** <hashAlgo>] [**-lifetime** <positive_integer>] [**-psk**] [**-publickey** <string>] show ipsec peer [**-name**] rm ipsec peer [**-name**] set ipsec parameter [**-encalgo** (AES | 3DES)] [**-hashalgo** <hashAlgo>] [**-lifetime** <positive_integer>] unset ipsec parameter [**-encalgo**] [**-hashalgo**] [**-lifetime**]

Deprecated Parameters

The following parameters are deprecated in release 9.3.

- add appfw profile **-stripcomments**
- set appfw profile **-stripcomments**
- add gslb vserver **-iptype**
- set gslb vserver **-iptype**
- unset lb vserver **-redirecturl**
- set ns tcpParam **-recvbuffsize**

New SNMP OIDs

The following table lists the new SNMP OIDs in release 9.3.

Table 1. New SNMP OIDs

OID	Description
ACL	
1.3.6.1.4.1.5951.4.1.1.22.6	Statistical information about the configured SimpleACL6s, in the rs9000 product family of NetScaler products.
1.3.6.1.4.1.5951.4.1.1.22.6.1	Total packets that matched a SimpleACL6 with action BRIDGE and were bridged by NetScaler.
1.3.6.1.4.1.5951.4.1.1.22.6.2	Packets dropped because they match a simple deny ACL6.
1.3.6.1.4.1.5951.4.1.1.22.6.3	Total packets that matched a SimpleACL6 with action ALLOW and got consumed by NetScaler.
1.3.6.1.4.1.5951.4.1.1.22.6.4	Packets matching a simple ACL6.
1.3.6.1.4.1.5951.4.1.1.22.6.5	Packets not matching any simple ACL6.
1.3.6.1.4.1.5951.4.1.1.22.6.6	Number of simple ACL6s configured.
Networking	
1.3.6.1.4.1.5951.4.1.1.24.1.25	VLAN alias name if configured.
1.3.6.1.4.1.5951.4.1.1.70	This table contains information about the IPv6 addresses configured on the NetScaler. This is indexed on nsInetAddressType and nsInetAddress.
1.3.6.1.4.1.5951.4.1.1.70.1.1	The address type of nsInetAddress.
1.3.6.1.4.1.5951.4.1.1.70.1.2	The IPv4/IPv6 address configured on the NetScaler.
1.3.6.1.4.1.5951.4.1.1.70.1.3	The netmask length.
1.3.6.1.4.1.5951.4.1.1.70.1.4	The IP address type.
1.3.6.1.4.1.5951.4.1.1.70.1.5	The IP address mode.
1.3.6.1.4.1.5951.4.1.1.70.1.6	The number of unused ports free on this IP address.
1.3.6.1.4.1.5951.4.1.1.70.1.7	The vLAN to which this IP address is bound.
1.3.6.1.4.1.5951.4.1.1.70.1.8	The bridge group to which this IP address is bound.

Application Firewall	
1.3.6.1.4.1.5951.4.1.1.64.1.32	Number of requests returning XML generic error from the backend server.
1.3.6.1.4.1.5951.4.1.1.64.1.33	Number of Signature violations seen by the Application Firewall.
1.3.6.1.4.1.5951.4.1.1.64.2.1.33	Number of requests returning XML generic violation from the backend server.
1.3.6.1.4.1.5951.4.1.1.64.2.1.34	Number of Signature violations seen by the Application Firewall.
Load Balancing	
1.3.6.1.4.1.5951.4.1.3.6.1.6	Number of times MSTS RDP Cookie got parsed on this vserver.
DNS	
1.3.6.1.4.1.5951.4.1.10.2.26	The name of the DNS key that is due for expiry.
1.3.6.1.4.1.5951.4.1.10.2.27	The amount of time for the key to expire.
1.3.6.1.4.1.5951.4.1.10.2.28	The unit of the time of expiry.
1.3.6.1.4.1.5951.4.1.10.2.29	The new name of the entity after the name was changed.
1.3.6.1.4.1.5951.4.1.10.2.30	The old name of the entity.

Deprecated SNMP OIDs

The following SNMP OIDs are deprecated in release 9.3.

- 1.3.6.1.4.1.5951.4.1.1.23.13
- 1.3.6.1.4.1.5951.4.1.1.63.2
- 1.3.6.1.4.1.5951.4.1.1.63.3
- 1.3.6.1.4.1.5951.4.1.1.63.5
- 1.3.6.1.4.1.5951.4.1.10.2.21
- 1.3.6.1.4.1.5951.4.1.10.2.22

Upgrading or Downgrading the System Software

NetScaler 9.3 offers new and updated features with increased functionality. A comprehensive list of enhancements is listed in the release notes accompanying the release announcement. Take a moment to read this document before you upgrade your software.

It is important to understand the licensing framework and types of licenses before you upgrade your software. A software edition upgrade may require new licenses, such as upgrading from the standard edition to the enterprise edition, the standard edition to the platinum edition, or the enterprise edition to the platinum edition.

Changes to the Licensing Framework

The licensing framework has changed since release 8.1, build 65.5. For more information, see http://support.citrix.com/article/ctx121062.

Citrix NetScaler Application Accelerator users have been upgraded to the Citrix Access Gateway Enterprise Edition without any change in functionality.

NetScaler Licenses

A NetScaler must be properly licensed before it can be deployed to distribute, optimize, or secure network traffic for Web applications. After you have obtained the licenses, you must install the licenses on your appliance, and then verify that the features corresponding to these licenses are enabled.

Obtaining NetScaler Licenses

The procedure to obtain NetScaler licenses has changed. For more information, see http://support.citrix.com/article/ctx121062.

Installing NetScaler Licenses

Install each license to use the feature controlled by that license.

To install the licenses by using the NetScaler command line

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials.
3. Switch to the shell prompt, create a license subdirectory in the nsconfig directory, if it does not exist, and copy the new license file(s) to this directory.

   Example

   ```
   login: nsroot
   Password: nsroot
   Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
   Done
   > shell
   Last login: Mon Aug  4 03:51:42 from 10.103.25.64
   root@ns# mkdir /nsconfig/license
   root@ns# cd /nsconfig/license
   ```

   Copy the new license file(s) to this directory.

To install the licenses by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as `http://192.168.100.1`.
2. In **User Name** and **Password**, type the administrator credentials.
3. In **Start in**, select **Configuration**, and then click **Login**, as shown in the following figure.

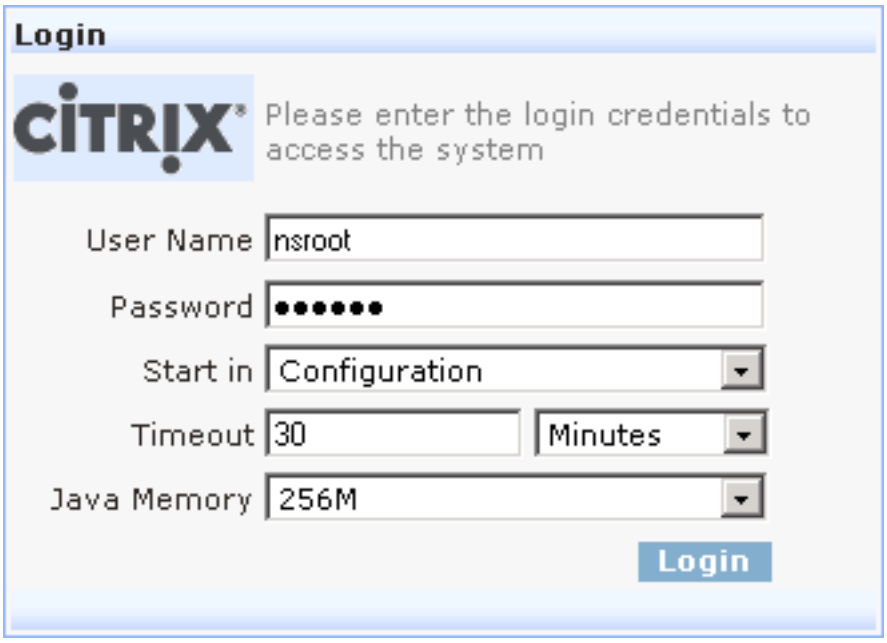


Figure 1. Login Screen

4. In the navigation pane, expand **System**, and then click **Licenses**.
5. In the **Licenses** pane, click **Manage Licenses**. If the /nsconfig/license directory does not exist, you are prompted to create it.
6. In the **Manage Licenses** dialog box, click **Add**.
7. In the **Select License Files** dialog box, navigate to the location of the license files and select the file you want to upload (for example, Citrix NetScaler IPv6 - Option.lic).
8. Click **Select**.
9. After your file is uploaded to the license directory, click **OK**.
10. When prompted to restart the NetScaler, do one of the following:
    - If you plan to upgrade your software, click **No**.
    - If you don’t plan to upgrade your software, click **Yes** to restart the NetScaler.

Verifying the Licensed Features

Before using a feature, make sure that your license supports the feature.

To verify the licensed features by using the NetScaler command line

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials.
3. At the command prompt, enter the sh ns license command to display the features supported by the license.

   Example

   ```
   sh ns license
         License status:
                        Web Logging: YES
                  Surge Protection: YES
                    .......

                    HTML Injection: YES
    Done
   ```

To verify the licensed features by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as `http://192.168.100.1`.

2. In **User Name** and **Password**, type the administrator credentials.

3. In **Start in**, select **Configuration**, and then click **Login**, as shown in the following figure.

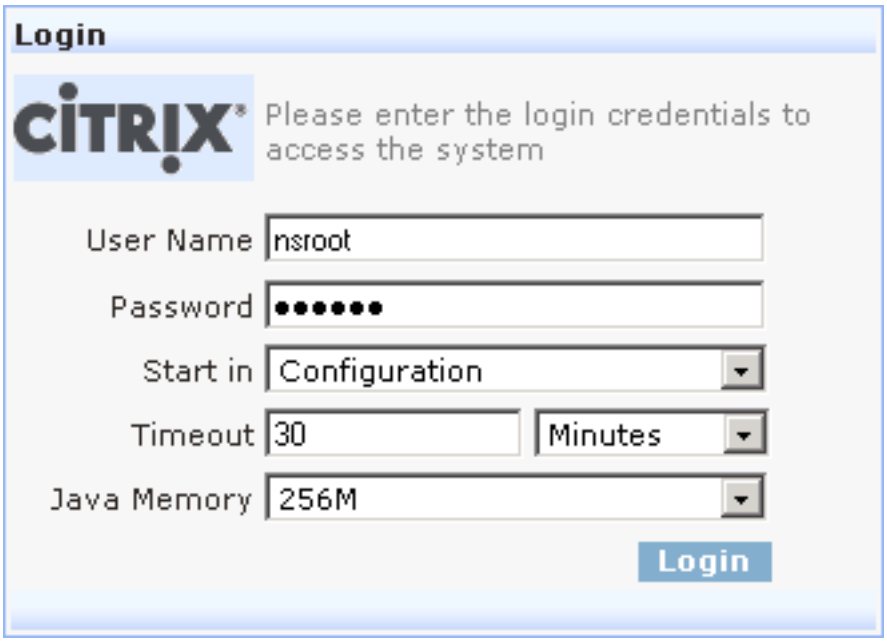


Figure 1. Login Screen

4. In the navigation pane, expand **System**, and then click **Licenses**. You will see a green check mark next to the licensed features.

Enabling or Disabling a Feature

When you use the NetScaler for the first time, you need to enable a feature before you can use its functionality. If you configure a feature before it is enabled, a warning message appears. The configuration is saved but it will apply only after the feature is enabled.

To enable a feature by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable a feature and verify the configuration:

- `enable feature <FeatureName>`
- `show feature`

Example

```
enable feature lb cs
done
 >show feature

      Feature                Acronym          Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               ON
3)    Load Balancing           LB               ON
4)    Content Switching        CS               ON
5)    Cache Redirection        CR               ON
.
.
.
24)   NetScaler Push           push             OFF
Done
```

The example shows how to enable load balancing (lb) and content switching (cs).
If the license key is not available for a particular feature, the following error message appears for that feature:

```
ERROR: feature(s) not licensed
```

Note: To enable an optional feature, you need a feature-specific license. For example, if you have purchased and installed the Citrix NetScaler Enterprise Edition license and need to enable the Integrated Caching feature, you first need to purchase and install the AppCache license.

To disable a feature by using the NetScaler command line

At the NetScaler command prompt, type the following commands to disable a feature and verify the configuration:

- `disable feature <FeatureName>`
- `show feature`

Example

```
> disable feature lb
 Done
> show feature

      Feature                 Acronym          Status
      -------                 -------          ------
1)    Web Logging               WL               OFF
2)    Surge Protection          SP               ON
3)    Load Balancing            LB               OFF
4)    Content Switching         CS               ON
.
.
.
24)   NetScaler Push            push             OFF
Done
>
```

The example shows how to disable load balancing (lb).

Access Gateway Universal License

The Access Gateway universal license limits the number of concurrent user sessions to the number of licenses purchased. If you purchase 100 licenses, you can have 100 concurrent sessions at any time. When a user ends a session, that license is released for the next user. A user who logs on to the Access Gateway from more than one computer occupies a license for each session.

If all licenses are occupied, no additional connections can be opened until a user ends a session or the administrator terminates the session using the configuration utility. When a connection is closed, the license is released and can be used for a new user.

Obtaining the Universal License

You need the following information before going to the Citrix Web site for the universal license.

The license code

You can find the code on the Access Gateway CD, in an email you receive from Citrix, or from the Subscription Advantage Management-Renewal-Information (SAMRI) system.

Your user ID and password for My Citrix

Register at My Citrix (www.mycitrix.com) to receive your user ID and password.

Note: If you cannot locate either the license code or your user ID and password, contact Citrix Customer Service.

The host name of the Access Gateway

The entry field for this name on My Citrix is case-sensitive, so make sure that you copy the host name exactly as it is configured on the NetScaler.

The number of licenses you want to include in the license file

You do not have to download all of the licenses you are entitled to at once. For example, if your company purchased 100 licenses, you can choose to download 50. You can allocate the rest in another license file at a later time. Multiple license files can be installed on the Access Gateway.

Note: Before obtaining your licenses, make sure you configure the host name of the NetScaler using the Setup Wizard and then restart the NetScaler.

To obtain your universal license

1. In a Web browser, go to http://www.citrix.com/ and click **My Citrix**.
2. Enter your user name and password. If this is the first time you are logging on to the site, you are asked for additional background information.
3. Under **My Tools**, point to **Choose a toolbox**, and click **Activation System/Manage Assets**.
4. In the **Current Tool** drop-down menu, **select Activate/Allocate** and follow the directions to obtain your license file.

Installing the Universal License

To install the license, see Installing NetScaler Licenses. After installation, verify that the license was installed correctly.

Verifying Installation of the Universal License

Before proceeding, verify that your universal license is installed correctly.

To verify installation of the universal license by using the NetScaler command line

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials.
3. Use the show license command to verify that “SSL VPN = YES” and that Maximum Users has increased from 5 to the expected number of concurrent users.

To verify installation of the universal license by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as `http://192.168.100.1`.
2. In **User Name** and **Password**, type the administrator credentials.
3. In the navigation pane, expand **System**, and then click **Licenses**.
4. In the Licenses pane, you will see a green check mark next to **Access Gateway**. The field **Maximum Access Gateway Users Allowed** displays the number of concurrent user sessions licensed on the NetScaler.

Upgrading to Release 9.3

You can upgrade to release 9.3 on a standalone NetScaler or a high availability (HA) pair by using the configuration utility or the NetScaler command line.

Important:

If an IPv6 address is configured as the NetScaler IP (NSIP) address, upgrading from release 8.1 to release 9.3 changes the NSIP address to the subnet IP (SNIP) address. To add the NSIP address after the upgrade, at the NetScaler command line, type:

```
rm ns ip6
```

```
add ns ip6 <ipv6 address> -type NSIP.
```

There is no change if the NSIP address is an IPv4 address.

Upgrading a Standalone NetScaler

Before upgrading the system software, make sure that you have the required licenses. For more information, see NetScaler Licenses. Software upgrades from 8.x to 9.x do not require a new license.

Note: When upgrading from release 8.0, 8.1, 9.0, 9.1, or 9.2, you have the option to use the configuration utility. All upgrades can be performed by using the command line, which is the recommended option. When using the upgrade wizard in the configuration utility to upgrade from release 8.0, do not use the **Device** option to upload your software.

To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, or 9.2 by using the command line

In the following procedure, <release> and <releasenumber> represent the release version you are upgrading to, and <targetbuildnumber> represents the build number that you are upgrading to. Refer to the table below for specific values.

Note: The 9.2 and 9.3 installation package name contains hyphens instead of underscores. The target build number for 9.2 and 9.3 classic contains _cl, for example build-9.2-25_cl.tgz and build-9.3-25_cl.tgz. The target build number for 9.2 and 9.3 nCore contains _nc, for example build-9.2-12_nc.tgz and build-9.3-12_nc.tgz. Older releases use the underscore as stated in the following procedure.

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.

2. Log on to the NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:`save config`

3. Create a copy of the ns.conf file. At the shell prompt, type:

   a. cd /nsconfig

   b. cp ns.conf ns.conf-.NS<releasenumber><currentbuildnumber>

   You should backup a copy of the configuration file on another computer.

4. Create a <releasenumber>nsinstall subdirectory in the /var/nsinstall directory.

5. Change directory to /var/nsinstall/<releasenumber>nsinstall, create a directory named build_<targetbuildnumber>, and change to this directory.

6. Download or copy the installation package (build_<release>_<build number>._nc.tgz) and the documentation bundle (ns-<releasenumber>-<build number>-doc.tgz) to this directory and extract the contents of the installation package. To download the installation package from the Citrix Web site, follow the steps below:

   a. Go to MyCitrix.com, log on using your credentials, and click **Support > Downloads**.

b. In the **Search Downloads by Product list**, select **Citrix NetScaler**.

c. Under **Firmware**, click the release and build number to download.

d. Click **Get Firmware**.

e. Click **Show Documentation** and then click **Get Documentation**.

7. Run the installns script to install the new version of the system software.

   Note: To install a FIPS appliance, run the installns script with the -F option. To automatically clean up the flash, run the installns script with the -c option.

   Warning: When upgrading to the NetScaler 9.3 nCore build, the installation script prompts you to delete the /var directory if the swap partition is smaller than 32 gigabytes (GB). If you receive this prompt, type N, save any important files located in /var to a backup location, and then re-run the installation script.

   If the free space available on the flash drive is insufficient to install the new build, the NetScaler prompts you to initiate a cleanup of the flash drive. For more information, see Auto Cleanup.

8. When prompted, restart the NetScaler.

Example

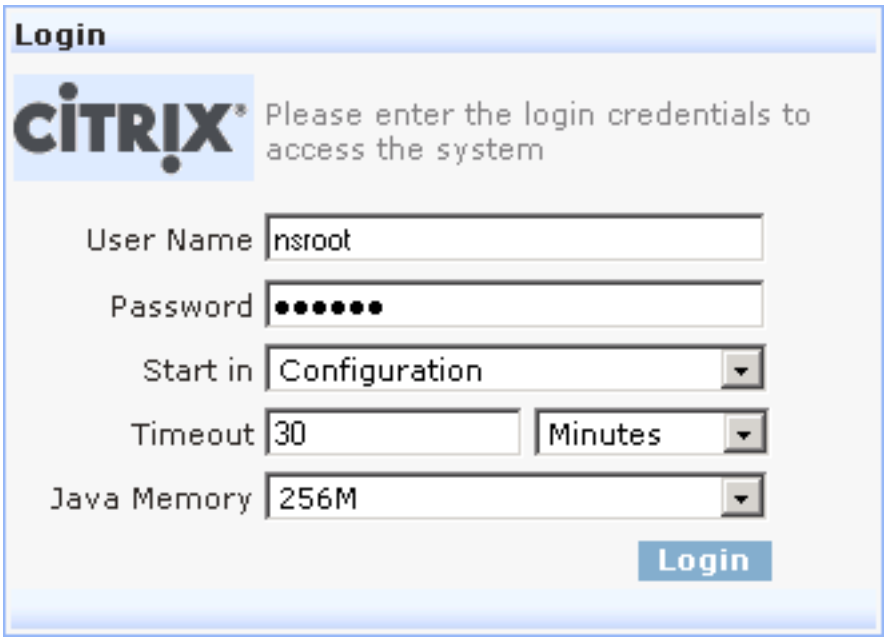
```
login: nsroot
Password: nsroot
Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
Done
> save config
> shell
Last login: Mon Aug  4 03:51:42 from 10.103.25.64
root@NSnnn# cd /nsconfig
root@NSnnn# cp ns.conf ns.conf.NS9.3-15
root@NSnnn# cd /var/nsinstall
root@NSnnn# cd 9.3nsinstall
root@NSnnn# mkdir build_25
root@NSnnn# cd build_25
root@NSnnn# ftp ... get build-9.3-25.tgz
root@NSnnn# get ns-9.3-25-doc.tgz
root@NSnnn# tar xzvf build-9.3-25.tgz
root@NSnnn# ./installns
installns version (9.3-25) kernel (ns-9.3-25.gz)
...
...
...
Copying ns-9.3-25.gz to /flash/ns-9.3-25.gz ...
```

```
Changing /flash/boot/loader.conf for ns-9.3-25 ...
Installing documentation...

Installation has completed.

Reboot NOW? [Y/N] Y
```

To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, or 9.2 by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as `http://10.102.29.50`.
2. In **User Name** and **Password**, type the administrator credentials.
3. 

   In **Start in**, select **Configuration**, and then click **Login**, as shown in the following figure.
4. In the configuration utility, in the navigation pane, click **System**.
5. In the **System Overview** page, click **Upgrade Wizard**.
6. Follow the instructions to upgrade the software.
7. When prompted, select **Reboot**.

   Note: After the upgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Upgrading a High Availability Pair

To upgrade the system software on NetScaler units in a high availability pair, you need to upgrade the software first on the secondary node and then on the primary node.

To upgrade NetScaler units in a high availability pair running release 8.0, 8.1, 9.0, 9.1, or 9.2 by using the NetScaler command line

Machine A is the primary node and machine B is the secondary node before the upgrade.

On machine B (original secondary node)

1. Follow the procedure for upgrading a standalone node as described in Upgrading a Standalone NetScaler.

2. After the NetScaler restarts, log on using the administrator credentials and enter the show ha node command to verify that the NetScaler is a secondary node and synchronization and propagation are disabled.

   Example

```
login: nsroot
Password: nsroot
Last login: Mon Aug  4 08:37:26 2008 from 10.102.29.9
Done
show ha node
        2 nodes:
1)      Node ID:      0
        IP:       10.0.4.2
        Node State: UP
        Master State: Secondary
        ...
        Sync State: AUTO DISABLED
        Propagation: AUTO DISABLED
        ...
Done
```

Note: Before upgrading the primary node (machine A), you have the option to test the new release by entering the force failover command on the secondary node (machine B). When you do so, machine B becomes the primary node. If machine B does not function as expected, enter the force failover command on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding. If machine B properly assumes the role of primary node, proceed with upgrading the former primary node (machine A).

On machine A (original primary node)

3. Follow the procedure for upgrading a standalone node as described in Upgrading a Standalone NetScaler.

4. After the NetScaler restarts, log on using the administrator credentials and enter the show ha node command to verify that the NetScaler is a secondary node and synchronization is disabled.

On machine B (new primary node)

5. Enter the show ha node command to verify whether machine B is the primary node.

On machine A (new secondary node)

6. Enter the show ns runningconfig command to verify whether the configuration of machine A has been synchronized with that of machine B

On machine B (new primary node)

7. Enter the save ns config command to save the configuration.

Machine B (original secondary node) is now the primary node and machine A (original primary node) is now the secondary node.

To upgrade NetScaler units in a high availability pair running release 8.0, 8.1, 9.0, 9.1, or 9.2 by using the configuration utility

1. Log on to the secondary node and perform the upgrade as described in To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, or 9.2 by using the configuration utility.

   Note: Before upgrading the primary node (machine A), you have the option to test the new release by entering the force failover command at the NetScaler command line on the secondary node (machine B). When you do so, machine B becomes the primary node. If machine B does not function as expected, enter the force failover command at the NetScaler command line on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding. If machine B properly assumes the role of primary node, proceed with upgrading the former primary node (machine A).

2. Log on to the primary node and perform the upgrade as described in To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, or 9.2 by using the configuration utility.

Upgrading to a Later Build within Release 9.3

You can upgrade from an earlier 9.3 build to a later 9.3 build on a standalone NetScaler or a high availability pair. This procedure can be performed by using the configuration utility or the NetScaler command line.

Upgrading a Standalone NetScaler to a Later Build

In the following procedure, <targetbuildnumber> is the build number that you are upgrading to within the 9.3 release.

To upgrade a standalone NetScaler running release 9.3 to a later build by using the NetScaler command line

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:

   save config
3. Create a copy of the ns.conf file. At the shell prompt, type:

   a. cd /nsconfig

   b. cp ns.conf ns.conf.NS9.3-<currentbuildnumber>
4. Change directory to /var/nsinstall/9.3nsinstall, create a directory named build_<targetbuildnumber>, and change to this directory.
5. Download or copy the installation package (build-9.3-<targetbuildnumber>_nc.tgz) and the documentation bundle (ns-9.3-<targetbuildnumber>-doc.tgz) to this directory and extract the contents of the installation package.
6. Run the **installns** script to install the new version of the system software.

   Note: To install a FIPS appliance, run the installns script with the -F option. To automatically clean up the flash, run the installns script with the -c option.

   If the free space available on the flash drive is insufficient to install the new build, the NetScaler prompts you to initiate a cleanup of the flash drive. For more information, see Auto Cleanup.
7. When prompted, restart the NetScaler.

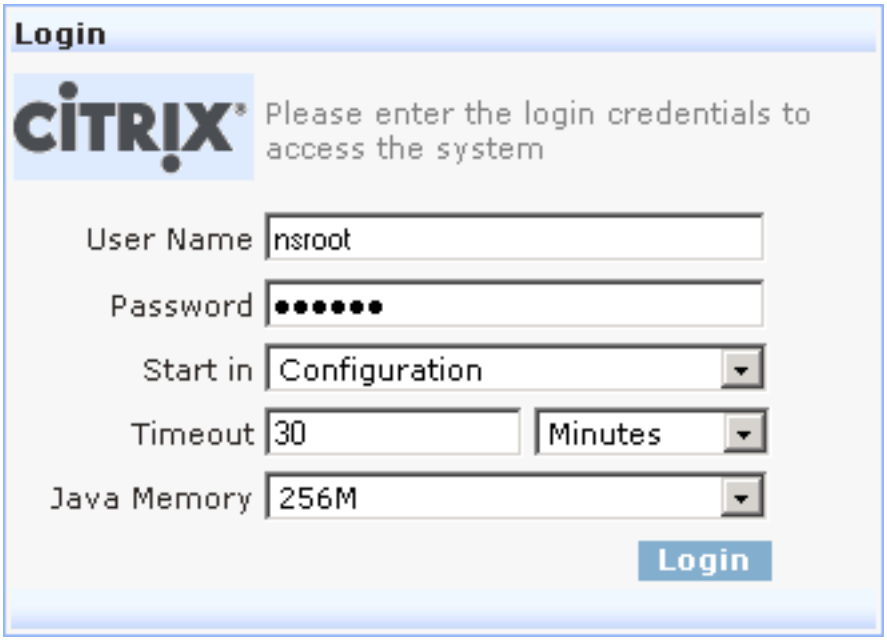
Example

```
login: nsroot
Password: nsroot
Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
Done
> save config
> shell
Last login: Mon Aug  4 03:51:42 from 10.103.25.64
root@NSnnn# cd /nsconfig
root@NSnnn# cp ns.conf ns.conf.NS9.3-15
root@NSnnn# cd /var/nsinstall
root@NSnnn# cd 9.3nsinstall
root@NSnnn# mkdir build_25
root@NSnnn# cd build_25
root@NSnnn# ftp ... get build-9.3-25.tgz
root@NSnnn# get ns-9.3-25-doc.tgz
root@NSnnn# tar xzvf build-9.3-25.tgz
root@NSnnn# ./installns
installns version (9.3-25) kernel (ns-9.3-25.gz)
...
...
...
Copying ns-9.3-25.gz to /flash/ns-9.3-25.gz ...
Changing /flash/boot/loader.conf for ns-9.3-25 ...
Installing documentation...

Installation has completed.

Reboot NOW? [Y/N] Y
```

To upgrade a standalone NetScaler running release 9.3 to a later build by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as `http://10.102.29.50`.
2. In **User Name** and **Password**, type the administrator credentials.
3. 

   In **Start in**, select **Configuration**, and then click **Login**, as shown in the following figure.
4. In the configuration utility, in the navigation pane, click **System**.
5. In the **System Overview** page, click **Upgrade Wizard**.
6. Follow the instructions to upgrade the software.
7. When prompted, select **Reboot**.

   Note: After the upgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Upgrading a NetScaler High Availability Pair to a Later Build

To upgrade the system software on NetScaler units in a high availability pair, you need to upgrade the software first on the secondary node and then on the primary node.

Warning: In certain rare cases, synchronization and propagation are disabled if you upgrade only one of the nodes in an HA pair to a later build.

To determine whether synchronization and propagation are disabled, at the NetScaler command line type:

show ha node

Note: In an HA setup, both nodes must run NetScaler nCore or NetScaler classic. If the nodes are running NetScaler classic and you want to migrate to NetScaler nCore of the same NetScaler release, propagation and synchronization are not supported during the migration process. Once migration is complete, you have to manually enable propagation and synchronization. The same applies if you migrate from NetScaler nCore to NetScaler classic.

If synchronization and propagation are disabled, a new command added on the new primary node will not be propagated to the new secondary node. Also, if you restart the new secondary node, it will not synchronize and fetch the running configuration from the new primary node, but it will use the configuration that was last saved before the node was restarted.

To resolve this situation, upgrade both of the nodes to the same build as soon as possible and make sure that no new command is added on the new primary node when a different build is running on the new secondary node.

In the following procedure, machine A is the original primary and machine B is the original secondary node, and <targetbuildnumber> is the build number that you are upgrading to within the 9.3 release.

To upgrade a NetScaler high availability pair to a later build

On machine B (original secondary node)

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:

   save config

3. Create a copy of the ns.conf file. At the shell prompt, type:

   a. cd **/nsconfig**

   b. cp **ns.conf ns.conf.NS9.3-**<currentbuildnumber>
4. Disable synchronization and propagation manually by entering the following commands in the order shown at the NetScaler command line:

   a. set HA node **-haSync** DISABLED

   b. set HA node **-haProp** DISABLED

   c. save config

   On machine A (original primary node)

5. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
6. Log on to the NetScaler by using the administrator credentials.
7. Disable synchronization and propagation manually by entering the following commands in the order shown at the NetScaler command line:

   a. set HA node **-haSync** DISABLED

   b. set HA node **-haProp** DISABLED

   c. save config

   On machine B (original secondary node)

8. Change directory to /var/nsinstall/9.3nsinstall, create a directory named build_<targetbuildnumber>, and change to this directory.
9. Download or copy the installation package (build-9.3-<targetbuildnumber>_nc.tgz) and the documentation bundle (ns-9.3-<targetbuildnumber>-doc.tgz) to this directory and extract the contents of the installation package.
10. Run the `installns` script to install the new version of the system software.

    Note: To install a FIPS appliance, run the installns script with the -F option. To automatically clean up the flash, run the installns script with the -c option.

    If the free space available on the flash drive is insufficient to install the new build, the NetScaler prompts you to initiate a cleanup of the flash drive. For more information, see Auto Cleanup.
11. When prompted, restart the NetScaler.
12. After the NetScaler restarts, log on using the administrator credentials and enter the show ha node command to verify that the NetScaler is a secondary node.

    Note: Before upgrading the primary node (machine A), you have the option to test the new build by entering the force failover command on the secondary node

(machine B). When you do so, machine B becomes the primary node. If machine B does not function as expected, enter the force failover command on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding. If machine B properly assumes the role of primary node, proceed with upgrading the former primary node (machine A) by following steps 13 through 19.

On machine A (original primary node)

13. Follow the procedure for upgrading a standalone node.
14. After the NetScaler restarts, log on using the administrator credentials and enter the show ha node command to verify that the NetScaler is a secondary node.
15. Enable synchronization and propagation manually by entering the following commands in the order shown at the NetScaler command line:
    a. set HA node **-haSync** ENABLED
    b. set HA node **-haProp** ENABLED
    c. save config

On machine B (new primary node)

16. Enter the show ha node command to verify that machine B is the primary node.
17. Enable synchronization and propagation by entering the following command in the order shown at the NetScaler command line:
    a. set HA node **-haSync** ENABLED
    b. set HA node **-haProp** ENABLED
    c. save config

On machine A (new secondary node)

18. Enter the show ns runningconfig command to verify that the configuration of machine A has been synchronized with that of machine B.

On machine B (new primary node)

19. Enter the save ns config command to save the current configuration.

Machine B (original secondary node) is now the primary node and machine A (original primary node) is now the secondary node.

Upgrading from the Classic Release to the nCore Release

To take advantage of the higher throughput and faster processing provided by nCore, you can upgrade your appliance from a classic build to an nCore build, either within the same release or from an earlier release to a later release.

Citrix nCore is supported by NetScaler MPX appliances and NetScaler VPX virtual appliances. Before you upgrade your appliance, make sure that your appliance meets the following requirements.

Following are the physical appliances that support nCore:

- MPX 5500
- MPX 7500/9500
- MPX 9700/10500/12500/15500
- MPX 9700/10500/12500/15500 10G
- MPX 9700/10500/12500/15500 10G FIPS
- MPX 11500/13500/14500/16500/18500/20500
- MPX 17500/19500/21500

If you are upgrading the software on the NetScaler VPX virtual appliance, make sure that that the following minimum memory and virtual CPU (VCPU) requirements are met.

- 2 GB RAM
- 2 VCPUs

For more information about upgrading to a later release, see Upgrading to release 9.3.

For more information about upgrading to a later build, see Upgrading to a Later Build within Release 9.3.

Downgrading from Release 9.3

You can downgrade to any release on a standalone NetScaler or a high availability pair by using the NetScaler command line.

Caution: Loss in configuration may occur when downgrading. You should compare the configurations before and after the downgrade, and then manually readd any missing entries.

This procedure provides steps to downgrade from release 9.3 to an earlier release. For downgrading to an earlier build within release 9.3, see Downgrading to an Earlier Build within Release 9.3.

Note: Downgrading using the configuration utility is not supported.

Downgrading a Standalone NetScaler

In the following procedure, <release> and <releasenumber> represent the release version you are downgrading to, and <targetbuildnumber> represents the build number that you are downgrading to. Refer to the table below for specific values.

Table 1. Release Version Values

Release Version	<release>	<releasenumber>
9.2	9.2	9.2
9.1	9.1	9.1
9.0	9.0	9.0
8.1	rhodes	8.1
8.0	andes	8.0
7.0	sierra	7.0

To downgrade a standalone NetScaler

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.

2. Log on to the NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:

   save config

3. Create a copy of the ns.conf file. At the shell prompt, type:

   a. cd **/nsconfig**

   b. cp **ns.conf ns.conf.NS9.3-**<currentbuildnumber>
4. Copy the <releasenumber> configuration file (ns.conf.NS<releasenumber>) to ns.conf. At the shell prompt, type:

   cp **ns.conf.NS**<releasenumber> **ns.conf**

   Note: ns.conf.NS<releasenumber> is the backup configuration file that is automatically created when the system software is upgraded from release version <releasenumber> to the current release version. There may be some loss in configuration when downgrading. After the appliance restarts, compare the configuration saved in step 3 with the running configuration, and make any adjustments for features and entities configured before the downgrade. Save the running configuration after making the changes.

   Important: If you are downgrading from release 9.3 to release 7.0 and routing is enabled, perform step 5. Otherwise, skip to step 6.

5. If routing is enabled, the ZebOS.conf file will contain the configuration. At the shell prompt, type:

   a. cd **/nsconfig**

   b. cp **ZebOS.conf ZebOS.conf.NS**9.3-<currentbuildnumber>

   c. cp **ZebOS.conf.NS7.0 ZebOS.conf**
6. Change directory to /var/nsinstall/<releasenumber>nsinstall, or create one if it does not exist.

7. Change directory to build_<targetbuildnumber>, or create one if it does not exist.

8. Download or copy the installation package (build_<release>_<targetbuildnumber>.tgz) and the documentation bundle (ns-<releasenumber>-<targetbuildnumber>-doc.tgz) to this directory and extract the contents of the installation package.

9. Run the `installns` script to install the new version of the system software. If the free space available on the flash drive is insufficient to install the new build, the NetScaler prompts you to initiate a cleanup of the flash drive. For more information, see Auto Cleanup.

10. When prompted, restart the NetScaler.

Example

```
login: nsroot
Password: nsroot
Last login: Tue Aug  5 01:38:25 2008 from 10.102.29.9
Done
> save config
> shell
Last login: Tue Aug  5 02:07:06 from 10.103.25.64
root@NSnnn# cd /nsconfig


root@NSnnn# cp ns.conf ns.conf.NS9.3-40
root@NSnnn# cp ns.conf.NS8.1 ns.conf
root@NSnnn# cd /var/nsinstall
root@NSnnn# mkdir 8.1nsinstall
root@NSnnn# cd 8.1nsinstall
root@NSnnn# mkdir build_58
root@NSnnn# cd build_58
root@NSnnn# ftp ... get build_81_58.tgz
root@NSnnn# get ns-8.1-58-doc.tgz
root@NSnnn# tar xzvf build_81_58.tgz
root@NSnnn# ./nsinstall
installns version (8.1-58) kernel (ns-8.1-58.gz)
...
...
...
Copying ns-8.1-58.gz to /flash/ns-8.1-58.gz ...
Changing /flash/boot/loader.conf for ns-8.1-58 ...
Installing documentation...

Installation has completed.

Reboot NOW? [Y/N] Y
```

Downgrading a High Availability Pair

To downgrade the system software on NetScaler units in a high availability pair, you need to downgrade the software first on the secondary node and then on the primary node. For instructions on downgrading each node separately, see Downgrading a Standalone NetScaler.

Downgrading to an Earlier Build within Release 9.3

You can downgrade from a later 9.3 build to an earlier 9.3 build on a standalone NetScaler or a high availability pair. This procedure must be performed by using the NetScaler command line.

Caution: Loss in configuration may occur when downgrading. You should compare the configurations before and after the downgrade, and then manually readd any missing entries.

Downgrading a Standalone NetScaler to an Earlier Build

In the procedure below, <targetbuildnumber> is the build number that you are downgrading to within the same release.

To downgrade a standalone NetScaler to an earlier build

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:

   save config
3. Create a copy of the ns.conf file. At the shell prompt, type:

   a. cd /nsconfig

   b. cp ns.conf ns.conf.NS9.3-<currentbuildnumber>
4. Copy ns.conf.NS9.3-<targetbuildnumber> to ns.conf, if it exists. At the shell prompt, type:

   cp ns.conf.NS9.3-<targetbuildnumber> ns.conf

   Caution: If ns.conf.NS9.3-<targetbuildnumber> does not exist, loss in configuration may occur when downgrading to an earlier build. The errors and warnings appear only on the console. Please watch the console closely for these errors and warnings. After the appliance restarts, compare the configuration saved in step 3 with the running configuration, and make any adjustments for features and entities configured before the downgrade. Save the running configuration after making the changes.
5. Change directory to /var/nsinstall/9.3nsinstall.
6. Change directory to build_<targetbuildnumber>, or create one if it does not exist.
7. Download or copy the installation package (build-9.3-<targetbuildnumber>_nc.tgz) and the documentation bundle (ns-9.3-<targetbuildnumber>-doc.tgz) to this directory and extract the contents of the installation package.
8. Run the installns script to install the old version of the system software. If the free space available on the flash drive is insufficient to install the new build, the NetScaler prompts you to initiate a cleanup of the flash drive. For more information, see Auto Cleanup.
9. When prompted, restart the NetScaler.

Example

```
login: nsroot
Password: nsroot
Last login: Tue Aug  5 01:38:25 2008 from 10.102.29.9
Done
> save config
> shell
Last login: Tue Aug  5 02:07:06 from 10.103.25.64
root@NSnnn# cd /nsconfig
root@NSnnn# cp ns.conf ns.conf.NS9.3-25
root@NSnnn# cp ns.conf.NS9.3-15 ns.conf
root@NSnnn# cd /var/nsinstall
root@NSnnn# cd 9.1nsinstall
root@NSnnn# cd build_15
root@NSnnn# ftp ... get build-9.3-15.tgz
root@NSnnn# get ns-9.3-15-doc.tgz
root@NSnnn# tar xzvf build-9.3-15.tgz
root@NSnnn# ./nsinstall
installns version (9.3-15) kernel (ns-9.3-15.gz)
...
...
...
Copying ns-9.3-15.gz to /flash/ns-9.3-15.gz ...
Changing /flash/boot/loader.conf for ns-9.3-15 ...
Installing documentation...

Installation has completed.

Reboot NOW? [Y/N] Y
```

Downgrading a NetScaler High Availability Pair to an Earlier Build

To downgrade the system software on NetScaler units in a high availability pair, you need to downgrade the software first on the secondary node and then on the primary node. For instructions on downgrading each node separately, see Downgrading a Standalone NetScaler to an Earlier Build.

Note: Note: In an HA setup, both nodes must run NetScaler nCore or NetScaler classic. If the nodes are running NetScaler classic and you want to migrate to NetScaler nCore of the same NetScaler release, propagation and synchronization are not supported during the migration process. Once migration is complete, you have to manually enable propagation and synchronization. The same applies if you migrate from NetScaler nCore to NetScaler classic.

Auto Cleanup

The cleanup procedure has been simplified in the later versions of release 7.0 (build 48 and later) and in releases 8.0, 8.1, 9.0, 9.1, and 9.2. You no longer have to manually delete build files from the flash drive. During the installation process, if the free space on the flash drive is found to be insufficient, the NetScaler prompts you to initiate the cleanup process.

Note: To automatically clean up the flash, run the installns script with the -c option.

When downgrading to release 7.0, the prompt looks like this:

```
Installation path for kernel will be /flash
Size of kernel ns-7.0-21.7.gz is 58003.323 kilobytes
Available space on /flash/ filesystem is 25075 kilobytes
Available space on /flash/ filesystem is insufficient to install ns-7.0-21.7.gz
Do you want Auto Cleanup [Y/N] ?
```

When upgrading to release 8.1, the prompt looks like this:

```
Installation path for kernel is /flash
Size of kernel ns-8.1-32.2.gz is 61062.235 kilobytes
Available space on /flash/ filesystem is 59108 kilobytes
Available space on /flash/ filesystem is insufficient to install ns-8.1-32.2.gz
Do you want installns to free space by archiving older releases? [Y/ N]
```

To initiate the cleanup process, press Y. Messages similar to the following appear:

```
Archiving older releases ...
  Creating the archive directory /var/nsbackup/ns_2007_2_16_1_6_26 ...
  Move //flash//ns-6.1-97.4.m.gz /var/nsbackup/ ns_2007_2_16_1_6_26ns-6.1-97.4.m.gz ...
  Move //flash//ns-8.1-32.2.gz /var/nsbackup/ns_2007_2_16_1_6_26ns-8.1-32.2.gz ...
Archive operation completed, free space is 156452, required space is 61062.235
```

The installation process automatically continues after successful completion of the cleanup.

Hardware Installation

The following sections describe the hardware installation and initial configuration for all NetScaler hardware platforms.

Introduction to the Hardware Platforms	Describes the NetScaler hardware platforms and provides detailed information about each platform and its components
Preparing for Installation	Describes how to unpack the NetScaler appliance and prepare the site and rack for installing the appliance. Lists the cautions and warnings that you should review before you install the appliance.
Installing the Hardware	Describes the steps to install the rails, mount the hardware, connect the cables, and turn on the appliance.
Initial Configuration	Describes how to perform initial configuration of your NetScaler appliance and assign management and network IP addresses.

Introduction to the Hardware Platforms

The NetScaler hardware platforms range from the single processor MPX 5500 and 7000 platforms to the high-capacity, fault-tolerant MPX 17500/19500/21500 hardware platforms. The various NetScaler hardware platforms are similar in that they use the same types of components, but different models provide different hardware capabilities. All NetScaler hardware platforms support the NetScaler operating system.

Some of the hardware platforms are available as dedicated application firewall appliances or secure application access appliances.

Introduction to the Hardware Platforms

The NetScaler hardware platforms range from the single processor MPX 5500 and 7000 platforms to the high-capacity, fault-tolerant MPX 17500/19500/21500 hardware platforms. The various NetScaler hardware platforms are similar in that they use the same types of components, but different models provide different hardware capabilities. All NetScaler hardware platforms support the NetScaler operating system.

Some of the hardware platforms are available as dedicated application firewall appliances or secure application access appliances.

Common Hardware Components

Each platform has front panel and back panel hardware components. The front panel has an LCD display and an RS232 serial console port. The number, type, and location of ports—copper Ethernet, copper and fiber SPF, SFP+, and XFP—vary by hardware platform. The back panel provides access to the power supply, fan, CompactFlash card, solid-state drive, and hard disk drive.

LCD Display

The LCD display on the front of every appliance displays messages about the current operating status of the appliance. These messages communicate whether your appliance has started properly and is operating normally. If the appliance is not operating normally, the LCD displays troubleshooting messages.

The LCD displays real-time statistics, diagnostic information, and active alerts. The dimensions of the LCD limit the display to two lines of 16 characters each, causing the displayed information to flow through a sequence of screens. Each screen shows information about a specific function.

The LCD has a neon backlight. Normally, the backlight glows steadily. When there is an active alert, it blinks rapidly. If the alert information exceeds the LCD screen size, the backlight blinks at the beginning of each display screen. When the appliance shuts down, the backlight remains on for one minute and then automatically turns off.

There are nine types of display screens on the LCD display. The first two screens in the following list, the booting screen and the startup screen, appear when your appliance is starting up. The other screens, except the out-of-service screen, can appear while the appliance is operating. They show configuration information, alerts, HTTP information, network traffic information, CPU load information, and port information for your appliance.

Booting Screen.

The booting screen is displayed immediately after the appliance is turned on. The first line displays the hardware platform, as shown in the following figure.

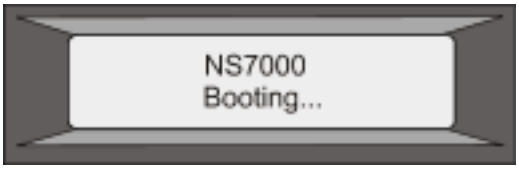


Figure 1. LCD Booting Screen

The newer MPX appliances display NSMPX followed by the platform number in the first line. For example, the MPX 7500/9500 appliances display NSMPX-7500. To view the model number, at the NetScaler command line, type show license. Scroll to the end of the command output to view the model number.

Startup Screen.

The startup screen is displayed for a few seconds after the appliance successfully begins operation. The first line displays the hardware platform, and the second line displays the software version and build number, as shown in the following figure.

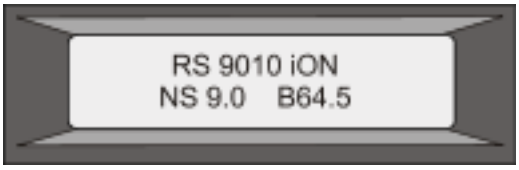


Figure 2. LCD Startup Screen

Out-of-Service Screen.

The out-of-service screen is displayed when the appliance has undergone a controlled shutdown, as shown in the following figure.

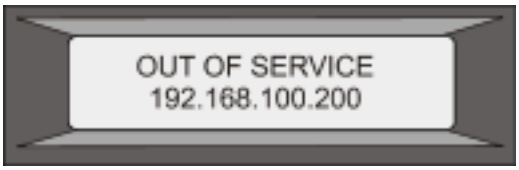


Figure 3. LCD Out-of-service Screen

Configuration Screen.

The first line displays the appliance status (STA, PRI, or SEC) and uptime. STA indicates that the appliance is in standalone mode, PRI indicates that the appliance is a primary node in a high availability (HA) pair, and SEC indicates that the appliance is a secondary node in an HA pair. Appliance uptime is displayed in HH:MM format. The second line displays the IP address of the appliance, as shown in the following figure.

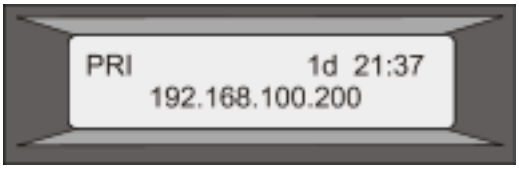


Figure 4. LCD Configuration Screen

Alert Screen.

An unknown alert is displayed differently than a known alert, as shown in the following figures. In either case, the first line displays the appliance status (STA, PRI, or SEC). STA indicates that the appliance is in standalone mode, PRI indicates that the appliance is a primary node in a high availability (HA) pair, and SEC indicates that the appliance is a secondary node in an HA pair. The second line displays the IP address of the appliance.

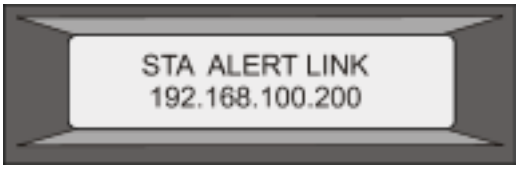


Figure 5. LCD Known Alert Screen

Figure 6. LCD Unknown Alert Screen

HTTP Statistics Screen.

The first line displays the rate of HTTP GETS per second. The second line displays the rate of HTTP POSTS per second, as shown in the following figure.

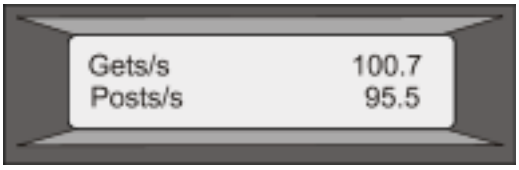


Figure 7. LCD HTTP Statistics Screen

Network Traffic Statistics Screen.

The first line displays the rate at which data is received, in megabits per second. The second line displays the rate of data transmission, in megabits per second, as shown in the following figure.

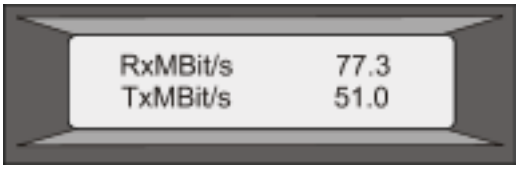


Figure 8. LCD Network Traffic Statistics Screen

CPU Load, Memory, and Connections Screen.

The first line displays CPU utilization and memory utilization as percentages. The second line displays the ratio of the number of server connections to the number of client connections.

Note: If the number of server or client connections exceeds 99,999, the number is displayed in thousands, indicated by the letter K.

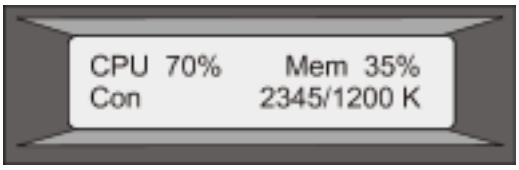


Figure 9. LCD CPU Load, Memory, and Connections Screen

Port Information Screen.

The S row displays port speed, flow control, and duplex information. The R row displays megabits received per second on the interface. The first port in each row is the management port.

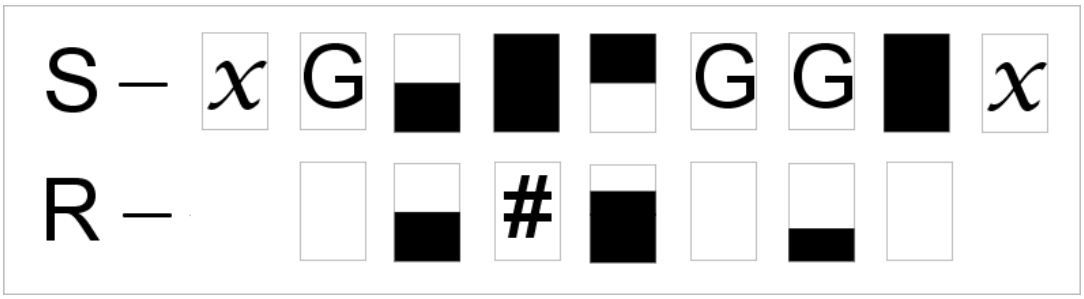


Figure 10. Port Information for an 8-port Appliance

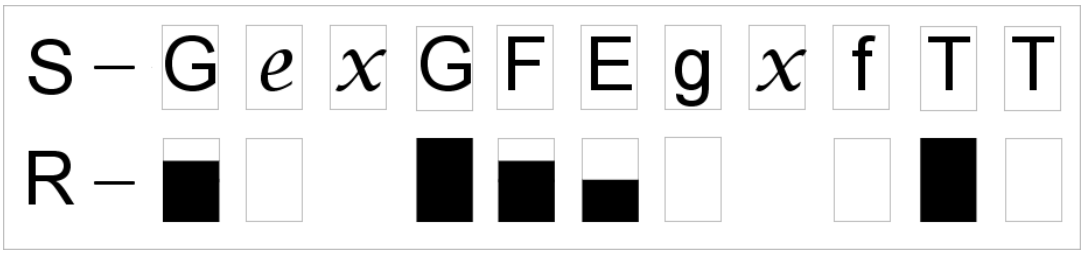


Figure 11. Port Information for a 10-port Appliance

The following table defines the various abbreviations and symbols that appear in the S row of the port information screen.

Table 1. Port Abbreviations and Symbols for S Row

S row abbreviation/symbol	Indicates

	A rate of 10 megabits per second, full duplex mode, and flow control OFF.
	A rate of 100 megabits per second, full duplex mode, and flow control OFF.
	A rate of 1 gigabit per second, full duplex mode, and flow control OFF.
	A rate of 10 gigabits per second, full duplex mode, and flow control OFF.
	A disconnected port. **Note:** The R row does not display an abbreviation or symbol for a disconnected port.
	Receive flow control regardless of speed or duplex mode.
	Transmit flow control regardless of speed or duplex mode.
	Receive and transmit flow control regardless of speed or duplex mode.

	A rate of 10 megabits per second, half duplex mode, and flow control OFF.
	A rate of 100 megabits per second, half duplex mode, and flow control OFF.
	A rate of 1 gigabit per second, half duplex mode, and flow control OFF.

The following table defines the various abbreviations and symbols that appear in the R row of the port information screen.

Table 2. Port Abbreviations and Symbols for R Row

R row abbreviation/symbol	Indicates
	The port is disabled.
	Receive speed is about 10% of line speed.
	Receive speed is about 50% of line speed.
	Receive speed is about 75% of line speed.
	Receive speed is about 100% of line speed.

Ports

Ports are used to connect the appliance to external devices. NetScaler appliances support RS232 serial ports, 10/100/1000Base-T copper Ethernet ports, 1-gigabit copper and fiber SFP ports, and 10-gigabit fiber SFP+ and XFP ports. All NetScaler appliances have a combination of some or all of these ports. For details on the type and number of ports available on your appliance, see the section describing that platform.

RS232 Serial Port

The RS232 serial console port on the front panel of each appliance provides a connection between the appliance and a computer, allowing direct access to the appliance for initial configuration or troubleshooting.

All hardware platforms ship with an appropriate serial cable used to connect your computer to the appliance. For instructions on connecting your computer to the appliance, see Installing the Hardware.

Copper Ethernet Ports

The copper Ethernet ports installed on many models of the appliance are standard RJ45 ports.

There are two types of copper Ethernet ports that may be installed on your appliance:

10/100BASE-T port

This type of port has a maximum transmission speed of 100 megabits per second (Mbps). Most platforms have at least one 10/100BASE-T port.

10/100/1000BASE-T port

This type of port has a maximum transmission speed of 1 gigabit per second, ten times faster than the other type of copper Ethernet port. Most platforms have at least one 10/100/1000Base-T port.

To connect any of these ports to your network, you plug one end of a standard Ethernet cable into the port and plug the other end into the appropriate network connector.

Management Ports

Management ports are standard copper Ethernet ports (RJ45) that are used for direct access to the appliance for system administration functions.

SFP, SFP+, and XFP Ports

An SFP port can operate at a speed of 1 Gbps. It accepts either a copper SFP transceiver for operation as a copper Ethernet port or a fiber SFP transceiver for operation as a fiber optic port.

The SFP+ and XFP ports are high-speed ports that can operate at speeds of 10 Gbps. You need a fiberoptic cable to connect to an SFP+ or XFP port. If the other end of the fiberoptic cable is attached to a 1GE SPF port, the 10GE SFP+ port automatically negotiates to match the speed of the 1GE SPF port.

The following tables list the maximum distance specifications for NetScaler pluggable media (SFP, SFP+, and XFP transceivers).

Note:

The tables are categorized by 1GE pluggable media and 10GE pluggable media.

The SFP+ modules are dual-speed capable and support both 1GE and 10GE, depending on the peer switch that the model connects to. These are listed in both tables.

Both tables have the following columns:

- SKU: Citrix maintains multiple SKUs for the same part.
- Description: The price list description of the part.
- Transmit Wavelength: The nominal transmit wavelength.
- Cable/Fiber Type: Fiber characteristics affect the maximum transmit distance achievable. This is especially true with 10GE on multi-mode fiber (MMF), where various dispersion components become dominant. For more information, see http://www.thefoa.org/tech/ref/basic/fiber.html.
- Typical Reach: Maximum transmit distance.
- Products: Some chassis are available with different media options. Use the appropriate data sheet to confirm that your particular chassis type supports the media.

1 GE Pluggable Media

The following table lists the maximum distance specifications for 1GE transceivers.

Table 1. Copper SFP (1GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Cable Type	Typical Reach (m)	Products

EW3A0000235, EW3B0000235, EW3C0000235, EW3D0000235, EW3E0000235, EW3F0000235, EW3P0000143, EW3X0000235, EW3Z0000087	Citrix NetScaler SFP Gigabit Ethernet Copper (100m) - **4 Pack**	n/a	Category 5 (Cat-5) Copper Cable	100 m	MPX 7500/9500, MPX 9700/10500/12500/15500, 12000, 10010, 9500, 9010, 9010 FIPS

Table 2. Short Reach Fiber SFP (1GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products
EW3A0000234, EW3B0000234, EW3C0000234, EW3D0000234, EW3E0000234, EW3F0000234, EW3P0000142, EW3X0000234, EW3Z0000086	Citrix NetScaler SFP Gigabit Ethernet SX (300m) - **4 Pack**	850nm (nominal)	50/125um MMF, 2000MHz-km (OM3)	550 m	MPX 7500/9500, MPX 9700/10500/12500/15500, 12000, 10010, 9500, 9010, 9010 FIPS
			50/125um MMF, 500MHz-km (OM2)	550 m	
			50/125um MMF, 400MHz-km	550 m	
			62.5/125um MMF, 200MHz-km (OM1)	300 m	
			62.5/125um MMF, 160MHz-km	300 m	

Table 3. Short Reach Fiber SFP (1GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products

EW3A0000710, EW3B0000710, EW3C0000710, EW3D0000710, EW3E0000710, EW3F0000710, EW3P0000557, EW3X0000710, EW3Z0000585	Citrix NetScaler SFP Gigabit Ethernet Short Range (300m) - **Single**	850nm (nominal)	50/125um MMF, 2000MHz-km (OM3)	550 m	MPX 9700/10500/12500/15500, MPX 17500/19500/21500, MPX 11500/13500/14500/16500/18500/20500
			50/125um MMF, 500MHz-km (OM2)	550 m	
			50/125um MMF, 400MHz-km	550 m	
			62.5/125um MMF, 200MHz-km (OM1)	275 m	
			62.5/125um MMF, 160MHz-km	220 m	

Table 4. Long Reach Fiber SFP (1GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products
EW3A0000712, EW3B0000712, EW3C0000712, EW3D0000712, EW3E0000712, EW3F0000712, EW3P0000559, EW3X0000712, EW3Z0000587	Citrix NetScaler SFP Gigabit Ethernet LX - **Single**	1310nm (nominal)	9/125um SMF	10 km	MPX 7500/9500, MPX 9700/10500/12500/15500, 12000, 10010, 9500, 9010, 9010 FIPS

Table 5. Long Reach Fiber SFP (1GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products
EW3A0000711, EW3B0000711, EW3C0000711, EW3D0000711, EW3E0000711, EW3F0000711, EW3P0000558, EW3X0000711, EW3Z0000586	Citrix NetScaler SFP Gigabit Ethernet Long Range (10km) - **Single**	1310nm (nominal)	9/125um SMF	10 km	MPX 9700/10500/12500/15500, MPX 17500/19500/21500, MPX 11500/13500/14500/16500/18500/20500

10 GE Pluggable Media

The following table lists the maximum distance specifications for 10GE transceivers.

Table 6. Short Reach Fiber SFP+ (10GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products
EW3A0000710, EW3B0000710, EW3C0000710, EW3D0000710, EW3E0000710, EW3F0000710, EW3P0000557, EW3X0000710, EW3Z0000585	Citrix NetScaler SFP+ 10 Gigabit Ethernet Short Range (300m) - **Single**	850nm (nominal)	50/125um MMF, 2000MHz-km (OM3)	300 m	MPX 9700/10500/12500/15500, MPX 17500/19500/21500, MPX 11500/13500/14500/16500/18500/20500, MPX 17550/19550/20550/21550
			50/125um MMF, 500MHz-km (OM2)	82 m	
			50/125um MMF, 400MHz-km	66 m	
			62.5/125um MMF, 200MHz-km (OM1)	33 m	
			62.5/125um MMF, 160MHz-km	26 m	

Table 7. Short Reach XFP (10GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products

EW3A0000713, EW3B0000713, EW3C0000713, EW3D0000713, EW3E0000713, EW3F0000713, EW3P0000560, EW3X0000713, EW3Z0000588	Citrix NetScaler XFP Short Range 10 Gigabit Ethernet(300m) - **Single**	850nm (nominal)	50/125um MMF, 2000MHz-km (OM3)	300 m	12000, MPX 15000/17000
			50/125um MMF, 500MHz-km (OM2)	82 m	
			50/125um MMF, 400MHz-km	66 m	
			62.5/125um MMF, 200MHz-km (OM1)	33 m	
			62.5/125um MMF, 160MHz-km	26 m	

Table 8. Long Reach Fiber SFP+ (10GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products
EW3A0000711, EW3B0000711, EW3C0000711, EW3D0000711, EW3E0000711, EW3F0000711, EW3P0000558, EW3X0000711, EW3Z0000586	Citrix NetScaler SFP+ 10 Gigabit Ethernet Long Range (10km) - **Single**	1310nm (nominal)	9/125um SMF	10 km	MPX 9700/10500/12500/15500, MPX 17500/19500/21500, MPX 11500/13500/14500/16500/18500/20500, MPX 17550/19550/20550/21550

Table 9. Long Reach Fiber XFP (10GE) Distance Specifications

SKU	Description	Transmitter Wavelength (nm)	Fiber Type	Typical Reach (m)	Products
EW3A0000714, EW3B0000714, EW3C0000714, EW3D0000714, EW3E0000714, EW3F0000714, EW3P0000561, EW3X0000714, EW3Z0000589	Citrix NetScaler XFP Long Range 10 Gigabit Ethernet(10 km) - **Single**	1310nm (nominal)	9/125um SMF	10 km	12000, MPX 15000/17000

LED Port-Status Indicators

Note: This section applies to the MPX 5500, MPX 7500/9500, MPX 9700/10500/12500/15500, MPX 17500/19500/21500, MPX 11500/13500/14500/16500/18500/20500, and MPX 17550/19550/20550/21550 appliances.

The port LEDs show whether the link is established and traffic is flowing through the port. The following table describes the LED indicators for each port. There are two LED indicators for each port type.

Table 10. LED port-status indicators

Port Type	LED Location	LED Function	LED Color	LED Indicates
SFP+ (10 Gbps)	Left	Link/ Activity	Off	No link.
			Solid green	Link is established but no traffic is passing through the port.
			Blinking green	Traffic is passing through the port.
	Right	Speed	Off	No connection.
			Solid green	Traffic rate of 10 gigabits per second.
SFP (1 Gbps)	Left	Link/ Activity	Off	No link.
			Solid green	Link is established but no traffic is passing through the port.
			Blinking green	Traffic is passing through the port.
	Right	Speed	Off	No connection.
			Yellow	Traffic rate of 1 gigabit per second.

Ethernet (RJ45)	Left	Speed	Off	No connection, or a traffic rate of 10 megabits per second (Mbps).
			Green	Traffic rate of 100 Mbps.
			Yellow	Traffic rate of 1 gigabit per second.
	Right	Link/ Activity	Off	No link.
			Solid green	Link is established but no traffic is passing through the port.
			Blinking green	Traffic is passing through the port.
Management (RJ45)	Left	Speed	Off	No connection, or a traffic rate of 10 megabits per second (Mbps).
			Green	Traffic rate of 100 Mbps.
			Amber	Traffic rate of 1 gigabit per second.
	Right	Link/ Activity	Off	No link.
			Solid yellow	Link is established but no traffic is passing through the port.
			Blinking yellow	Traffic is passing through the port.

Power Supply

Appliances are configured with either a single power supply or, for higher capacity fault tolerant models, a dual power supply configuration.

The appliance ships with a standard power cord that plugs into the appliance's power supply and an NEMA 5-15 plug on the other end for connecting to the power outlet on the rack or in the wall.

For power-supply specifications, see Hardware Platforms, which describes the various platforms and includes a table summarizing the hardware specifications.

Note: If you suspect that a power-supply fan is not working, please see the description of your platform. On some platforms, what appears to be the fan does not turn, and the actual fan turns only when necessary.

For each power supply, a bi-color LED indicator shows the condition of the power supply.

Table 1. LED Power Supply Indicators

Power Supply Type	LED Color	LED Indicates
AC	OFF	No power to any power supply.
	Flashing RED	No power to this power supply.
	Flashing GREEN	Power supply is in standby mode.
	GREEN	Power supply is functional.
	RED	Power supply failure.
DC	OFF	No power to any power supply.
	Flashing RED	No power to this power supply.
	Flashing BLUE	Power supply is in standby mode.
	BLUE	Power supply is functional.
	RED	Power supply failure.

CompactFlash Card

The CompactFlash card contains the operating system for all hardware platforms except for the MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances. CompactFlash is mounted as /flash.

Solid-State Drive

The solid-state drive on the MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances contains the operating system. It is mounted as /flash.

Hard Disk Drive

The hard disk drive contains logs and other data files on all hardware platforms. It is mounted as /var.

Hardware Platforms

The various NetScaler hardware platforms offer a wide range of features, communication ports, and processing capacities. All the MPX platforms have multicore processors.

Citrix NetScaler 7000

The Citrix NetScaler 7000 appliance is a 1U appliance, with 1 single-core processor, and 1 gigabyte (GB) of memory.

Note: NetScaler 9.3 nCore release is not supported on this hardware platform.

The following figure shows the front panel of the 7000.

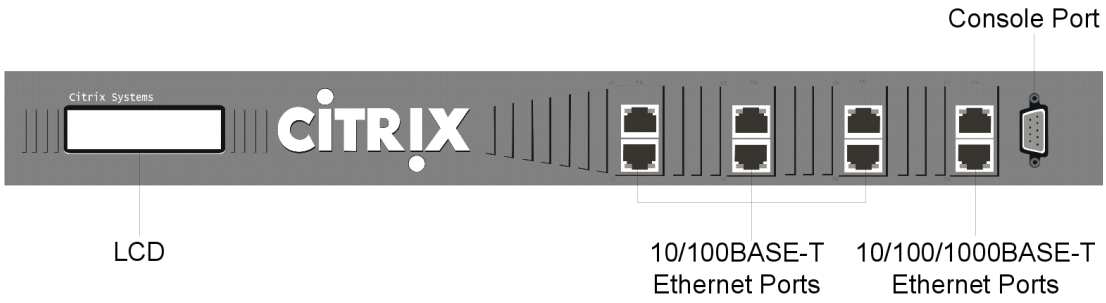


Figure 1. Citrix NetScaler 7000, front panel

The 7000 has the following ports:

- Six 10/100BASE-T copper Ethernet ports, named 1/1, 1/2, 1/3, 1/4, 1/5, and 1/6. Port 1/1 is the upper-left port, port 1/2 is the port beneath it, and the other 10/100BASE-T ports are named sequentially as you move from left to right, top to bottom.
- Two 10/100/1000BASE-T copper Ethernet ports, named 1/7 and 1/8. Port 1/7 is the upper port, and port 1/8 is the port beneath it.
- RS232 serial Console Port.

Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

The following figure shows the back panel of the 7000.

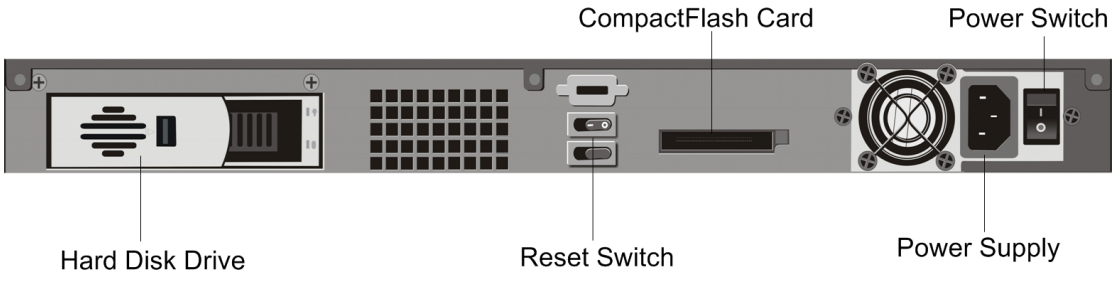


Figure 2. Citrix NetScaler 7000, back panel

The following components are visible on the back panel of the 7000:

- Removable Hard Disk Drive that is used to store user data.
- Appliance Reset Switch, which signals the 7000 to perform an orderly shutdown after saving all data.
- Removable CompactFlash Card that is used to store the operating system.
- Power Switch, which turns off power to the 7000 just as if you were to unplug it.
- Power Supply rated at 250 watts, 110-220 volts.

Citrix NetScaler 9010

The Citrix NetScaler 9010 appliance is a 2U appliance, with 1 single-core processor, and 2 GB of memory.

Note: NetScaler 9.3 nCore release is not supported on this hardware platform.

There are three models of the 9010: the copper Ethernet version, the fiber SFP (Small Form Factor Pluggable) version, and the FIPS (Federal Information Processing Standards) version. The following figure shows the front panel of the 9010 model with copper Ethernet ports.

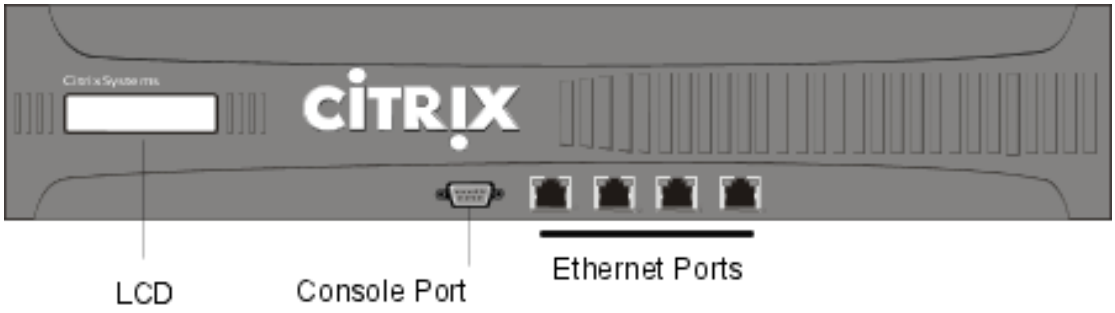


Figure 1. Citrix NetScaler 9010 front panel, with copper Ethernet ports

The following figure shows the front panel of the 9010 with fiber SFP ports.

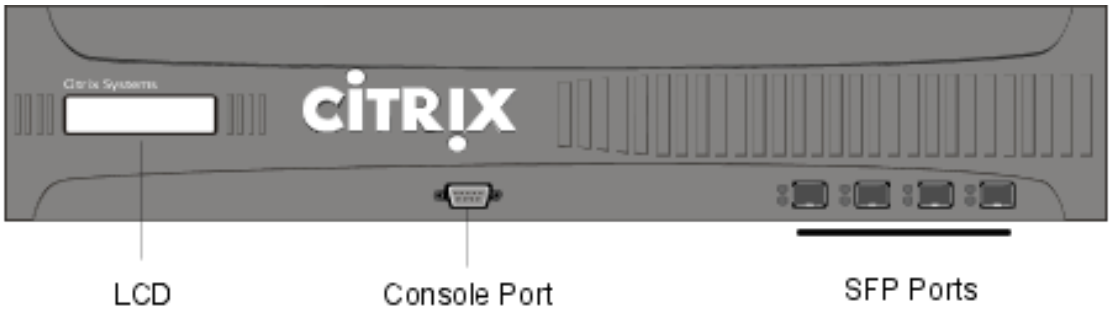


Figure 2. Citrix NetScaler 9010 front panel, with SFP ports

Depending on the model, the following components are visible on the front panel of the 9010:

- RS232 serial Console Port.
- Network ports
    - 9010, copper Ethernet model. Four copper Ethernet 10/100/1000BASE-T ports, numbered 1/1, 1/2, 1/3, and 1/4 from left to right.

- 9010, SFP model. Four SFP ports, numbered 1/1, 1/2, 1/3, and 1/4 from left to right. When facing the bezel, the upper LEDs to the left of each optical SFP port inset represent connectivity. They are lit and amber in color when active. The lower LEDs represent throughput. They are lit and green when active.

- 9010 FIPS model. Four ports, numbered 1/1, 1/2, 1/3, and 1/4 from left to right.

Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

The following figure shows the back panel of the 9010 models.

Note: The back panels of the three 9010 models are the same.

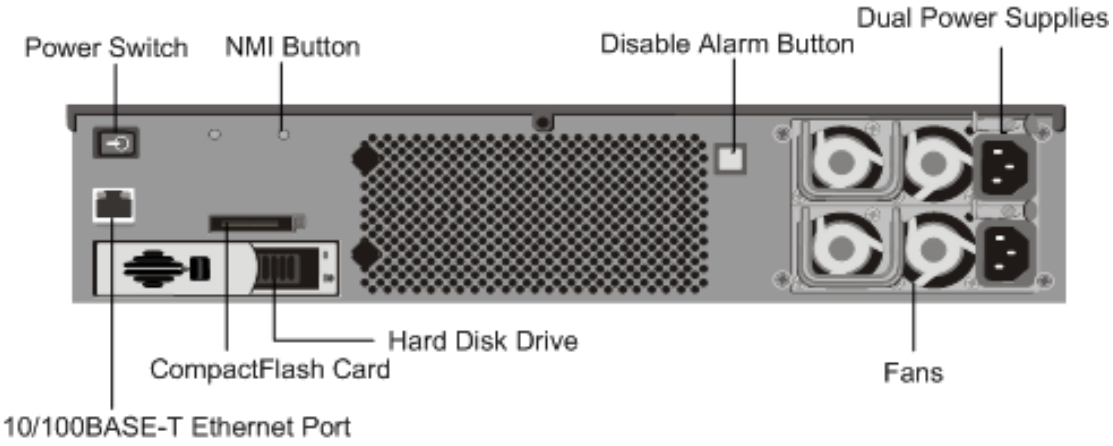


Figure 3. Citrix NetScaler 9010, back panel

The following components are visible on the back panel of the 9010 models:

- Power Switch, which turns off power to the 9010, just as if you were to unplug both power supplies.

- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the NetScaler. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.

- Disable Alarm Button, which silences the alarm that the 9010 sounds when it is receiving power from only one of its power supplies. Press this button to prevent the power alarm from sounding when you have plugged the 9010 into only one power outlet or when one power supply is malfunctioning and you wish to continue operating the 9010 until it is repaired.

- Dual Power Supplies, each rated at 500 watts, 110-220 volts. You plug separate power cords into the power supplies and connect them to separate wall sockets. The 9010 functions properly with a single power supply; the extra power supply serves as a backup.

- 10/100BASE-T copper Ethernet port, numbered 0/1.

- Removable CompactFlash Card that is used to store the operating system.
- Removable Hard Disk Drive that is used to store user data.

Citrix NetScaler 10010

The Citrix NetScaler 10010 appliance is a 2U appliance, with 1 single-core processor, and 4 GB of memory.

Note: NetScaler 9.3 nCore release is not supported on this hardware platform.

The following figure shows the front panel of the 10010.

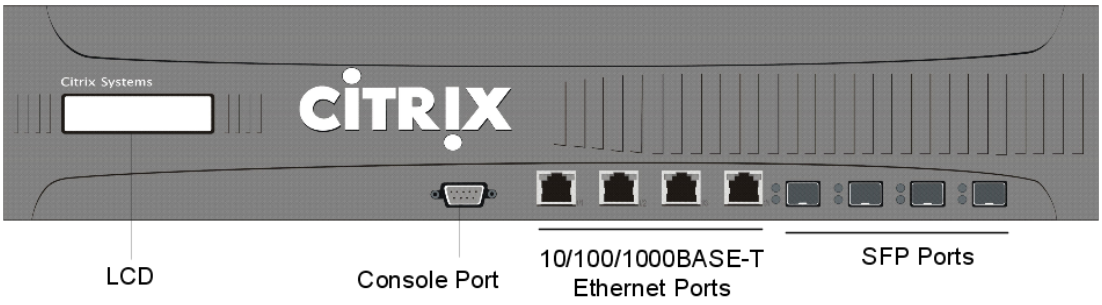


Figure 1. Citrix NetScaler 10010, front panel

Depending on the model, the following components are visible on the front panel of the 10010:

- RS232 serial Console Port.
- Four copper Ethernet 10/100/1000BASE-T ports, numbered 1/5, 1/6, 1/7, and 1/8 from left to right.
- Four Small Form Factor Pluggable (SFP) ports, numbered 1/1, 1/2, 1/3, and 1/4 from left to right. When facing the bezel, the upper LEDs to the left of each optical SFP port inset represent connectivity. They are lit and amber in color when active. The lower LEDs represent throughput. They are lit and green when active.

Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

The following figure shows the back panel of the 10010.

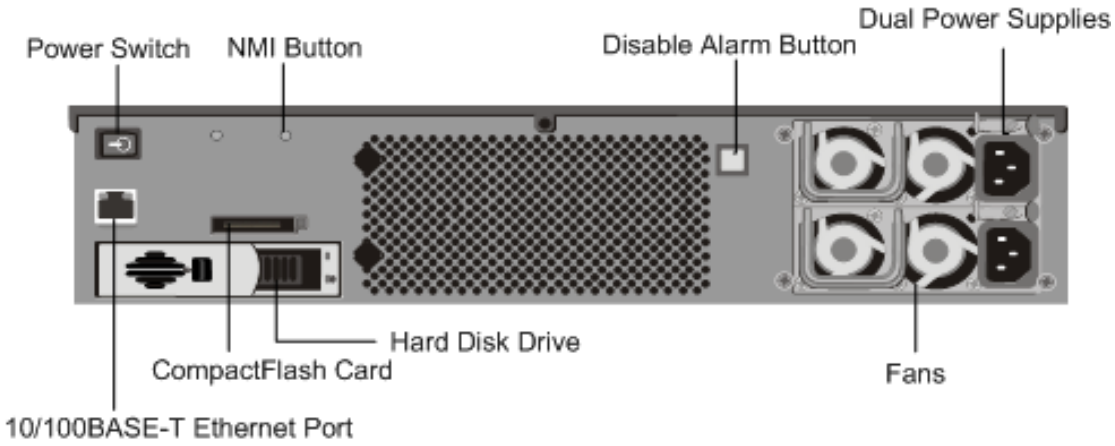


Figure 2. Citrix NetScaler 10010, back panel

The following components are visible on the back panel of the 10010:

- Power Switch, which turns off power to the 10010, just as if you were to unplug both power supplies.
- Non-maskable interrupt (NMI) button, which signals the 10010 to perform an orderly shutdown after saving all files. You must use a pen, pencil, or other pointed object to press this button, which is located inside a small hole to prevent it being pressed accidentally.
- Disable Alarm Button, which silences the alarm that the 10010 sounds when it is receiving power from only one of its power supplies. Press this button to prevent the power alarm from sounding when you have plugged the 10010 into only one power outlet or when one power supply is malfunctioning and you wish to continue operating the 10010 until it is repaired.
- Dual Power Supplies, each rated at 500 watts, 110-220 volts. You plug separate power cords into the power supplies and connect them to separate wall sockets. The 10010 functions properly with a single power supply; the extra power supply serves as a backup.
- 10/100BASE-T copper Ethernet port, numbered 0/1.
- Removable CompactFlash Card that is used to store the operating system.
- Removable Hard Disk Drive that is used to store user data.

Citrix NetScaler 12000

The Citrix NetScaler 12000 appliance is a 2U appliance, with 2 single-core processors, and 4 GB of memory. The 12000 is a high-capacity, fault-tolerant hardware platform intended for heavy use in data center environments.

Note: NetScaler 9.3 nCore release is not supported on this hardware platform.

The 12000 comes in four models: the fiber model, the copper model, the mixed model, and the 10G model. The following figure shows the front panel of the 12000 fiber model.

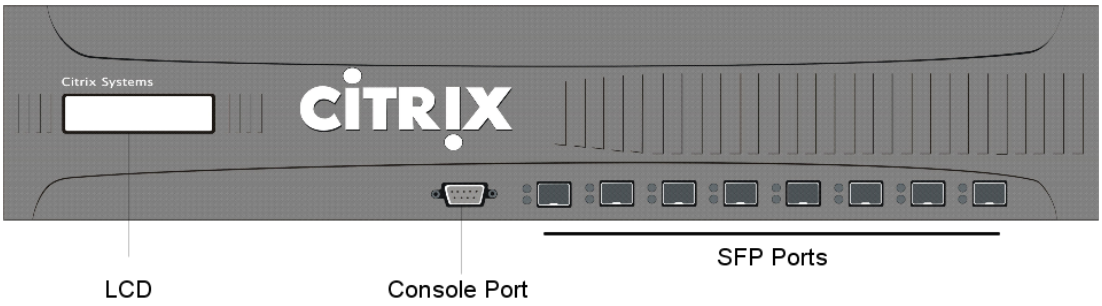


Figure 1. Citrix NetScaler 12000 fiber, front panel

The fiber model has eight fiber Small Form Factor Pluggable (SPF) ports. The copper model has eight copper SFP ports instead of eight fiber SFP ports, located in the same places. The mixed model has four copper SFP ports located in the left four positions and four fiber SFP ports located in the right four positions.

The following figure shows the front panel of the 12000-10G model.

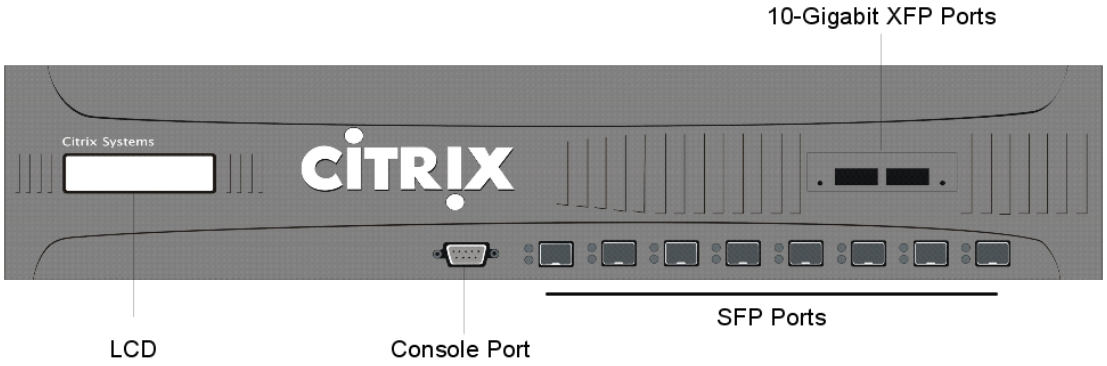


Figure 2. Citrix 12000-10G, front panel

Depending on the model, the following components are visible on the front panel of the 12000:

- RS232 serial Console Port.

- Network ports
  - 12000 Fiber. Eight fiber SFP ports, numbered 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, and 1/8 from left to right.
  - 12000 Copper. Eight copper SFP ports, numbered 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, and 1/8 from left to right.
  - 12000 Mixed. Four copper SFP ports, numbered 1/1, 1/2, 1/3, and 1/4, and four fiber SFP ports, numbered 1/5, 1/6, 1/7, and 1/8 from left to right.
  - 12000-10G. Eight SFP ports, numbered 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, and 1/8 from left to right, and two XFP (10-Gigabit Small Form Factor Pluggable) ports, numbered 1/9 and 1/10. When facing the bezel, the upper LEDs to the left of each optical SFP port represent connectivity. They are lit and amber in color when active. The lower LEDs represent throughput. They are lit and green when active.

  Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

The following figure shows the back panel of all 12000 models.

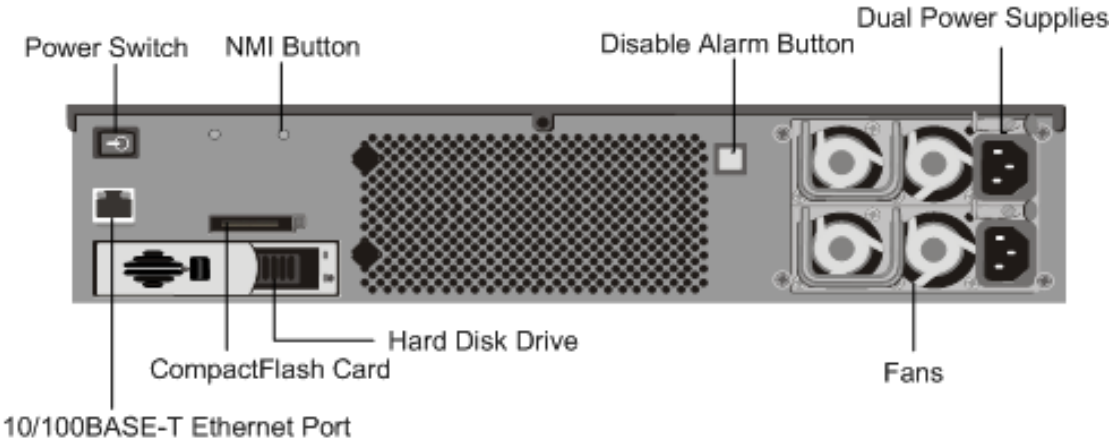


Figure 3. Citrix NetScaler 12000, back panel

The following components are visible on the back panel of the 12000:

- Power switch, which turns off power to the 12000, just as if you were to unplug both power supplies.
- Non-maskable interrupt (NMI) button, which signals the 12000 to perform an orderly shutdown after saving all files. You must use a pen, pencil, or other pointed object to press this button, which is located inside a small hole to prevent it being pressed accidentally.
- Disable Alarm Button, which silences the alarm that the 12000 sounds when it is receiving power from only one of its power supplies. Press this button to prevent the power alarm from sounding when you have plugged the 12000 into only one power outlet or when one power supply is malfunctioning and you wish to continue operating the 12000 until it is repaired.

- Dual Power Supplies, each rated at 500 watts, 110-220 volts. You plug separate power cords into the power supplies and connect them to separate wall sockets. The 12000 functions properly with a single power supply; the extra power supply serves as a backup.
- 10/100BASE-T copper Ethernet port, numbered 0/1.
- Removable CompactFlash Card that is used to store the operating system.
- Removable Hard Disk Drive that is used to store user data.

Citrix NetScaler MPX 5500

The Citrix NetScaler MPX 5500 is a 1U appliance, with 1 dual-core processor, and 4 gigabytes (GB) of memory.

The following figure shows the front panel of the MPX 5500.

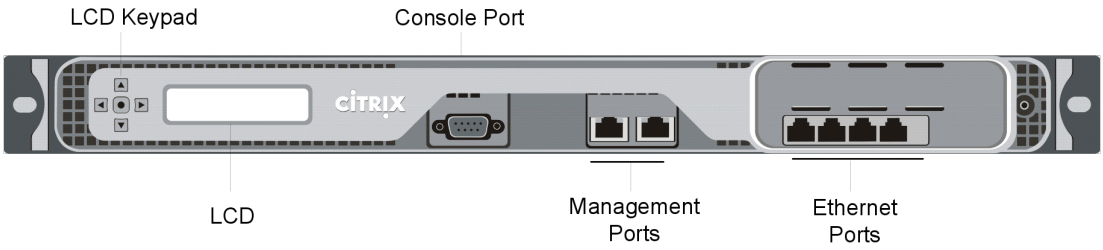


Figure 1. Citrix NetScaler MPX 5500, front panel

The MPX 5500 has the following ports:

- RS232 serial Console Port.
- Two 10/100/1000Base-T copper Ethernet Management Ports, numbered 0/1 and 0/2 from left to right. You can use these ports to connect directly to the appliance for system administration functions.
- Four 10/100/1000Base-T copper Ethernet ports numbered 1/1, 1/2, 1/3, and 1/4 from left to right.

Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

The following figure shows the back panel of the MPX 5500.

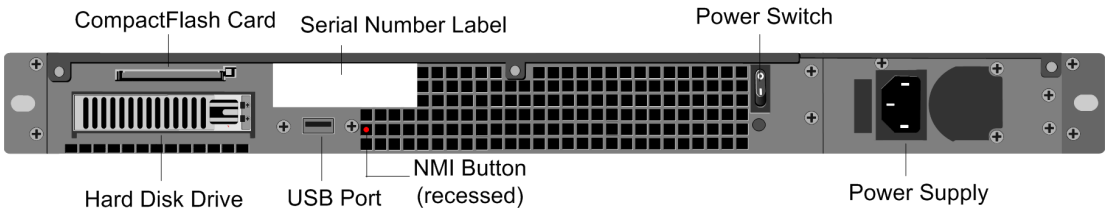


Figure 2. Citrix NetScaler MPX 5500, back panel

The following components are visible on the back panel of the MPX 5500:

- Four GB removable CompactFlash Card that is used to store the operating system.

- Power Switch, which turns off power to the MPX 5500, just as if you were to unplug the power supply. Press the switch for five seconds to turn off the power.
- Removable Hard Disk Drive that is used to store user data.
- USB port (reserved for a future release).
- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the NetScaler. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.
- Power supply rated at 300 watts, 110-220 volts. The power-supply fan is designed to turn on only when the internal temperature of the power supply reaches a certain value. You cannot see the fan turning on the back panel. What you can see is the fixed part of the fan that holds the spinning motor.

Citrix NetScaler MPX 7500 and MPX 9500

The Citrix NetScaler MPX 7500/9500 are 1U appliances, each with 1 quad-core processor, and 8 gigabytes (GB) of memory. The MPX 7500/9500 appliances are available in two port configurations: 8xCopper Ethernet (Cu) and 4xSFP+4xCu.

The following figure shows the front panel of the MPX 7500/9500 (8xCu) appliances.

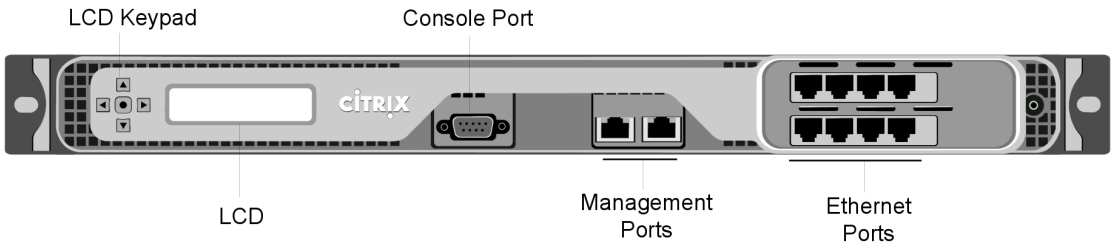


Figure 1. Citrix NetScaler MPX 7500/9500 (8xCu), front panel

The following figure shows the front panel of the MPX 7500/9500 (4xSFP+4xCu) appliances.

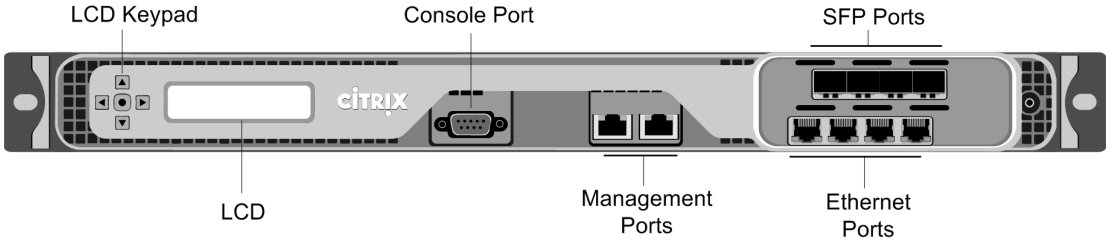


Figure 2. Citrix NetScaler MPX 7500/9500 (4xSFP+4xCu), front panel

Depending on the model, the appliance has the following ports:

- RS232 serial Console Port.
- Two 10/100/1000Base-T copper Ethernet management ports, numbered 0/1 and 0/2 from left to right. These ports are used to connect directly to the appliance for system administration functions.
- Network Ports
    - MPX 7500/9500 (8xCu). Eight 10/100/1000Base-T copper Ethernet ports numbered 1/1, 1/2, 1/3, and 1/4 on the top row from left to right, and 1/5, 1/6, 1/7, and 1/8 on the bottom row from left to right.

- MPX 7500/9500 (4xSFP+4xCu). Four 1-gigabit copper or fiber SFP ports numbered 1/1, 1/2, 1/3, and 1/4 on the top row from left to right, and four 10/100/1000BASE-T copper Ethernet Ports (RJ45) numbered 1/5, 1/6, 1/7, and 1/8 on the bottom row from left to right.

The following figure shows the back panel of the MPX 7500/9500appliance.

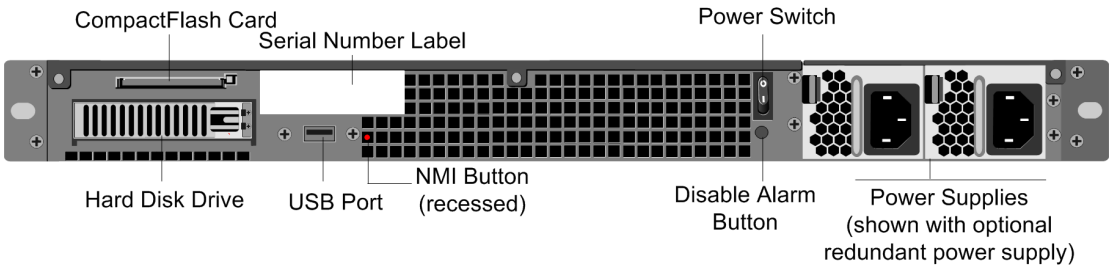


Figure 3. Citrix NetScaler MPX 7500/9500, back panel

The following components are visible on the back panel of the MPX 7500/9500:

- Four-gigabyte removable compact flash that is used to store the operating system.
- Power Switch, which turns off power to the MPX 7500/9500, just as if you were to unplug the power supply. Press the switch for five seconds to turn off the power.
- Removable Hard Disk Drive that is used to store user data.
- USB port (reserved for a future release).
- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the appliance. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.
- Disable Alarm Button. This button is functional only when the appliance has two power supplies.

  Press this button to stop the power alarm from sounding when you have plugged the MPX 7500/9500 into only one power outlet or when one power supply is malfunctioning and you want to continue operating the MPX 7500/9500 until it is repaired.

Citrix NetScaler MPX 9700, MPX 10500, MPX 12500, and MPX 15500

The Citrix NetScaler MPX 9700/10500/12500/15500 are 2U appliances, each with 2 quad-core processors, and 16 gigabytes (GB) of memory. All these appliances are also available in a 10G model and a FIPS model.

The following figure shows the front panel of the MPX 9700/10500/12500/15500.

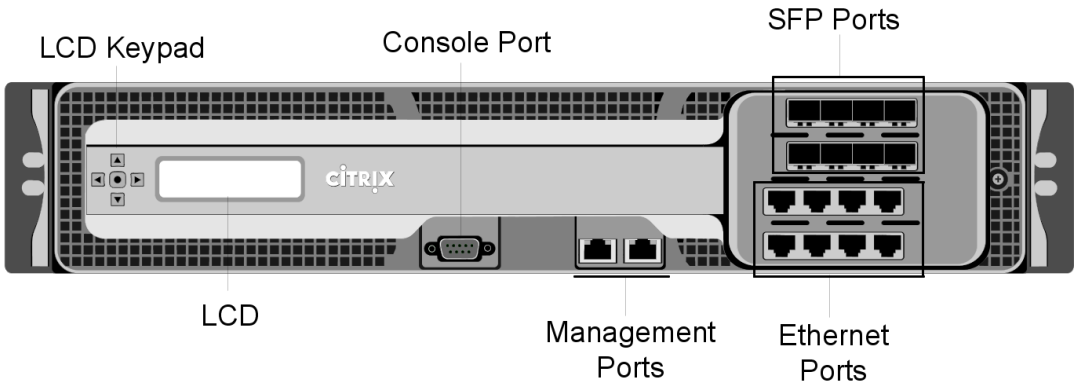


Figure 1. Citrix NetScaler MPX 9700/10500/12500/15500, front panel

The following figure shows the front panel of the MPX 9700/10500/12500/15500 10G.

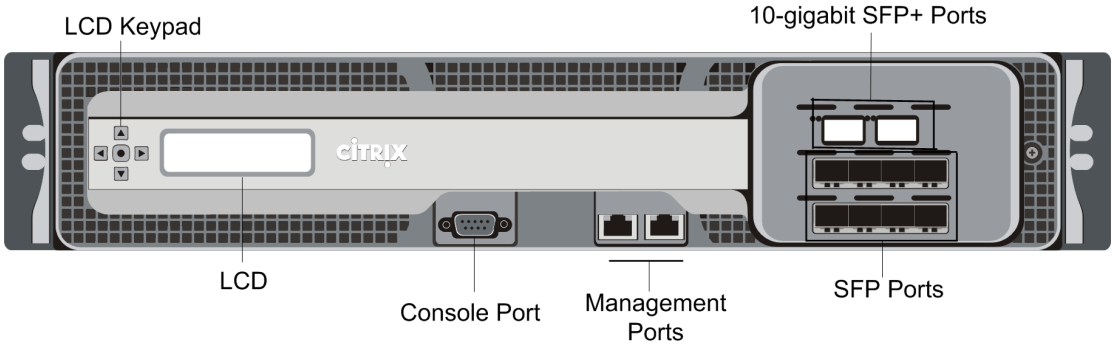


Figure 2. Citrix NetScaler MPX 9700/10500/12500/15500 10G, front panel

The following figure shows the front panel of the MPX 9700/10500/12500/15500 FIPS.

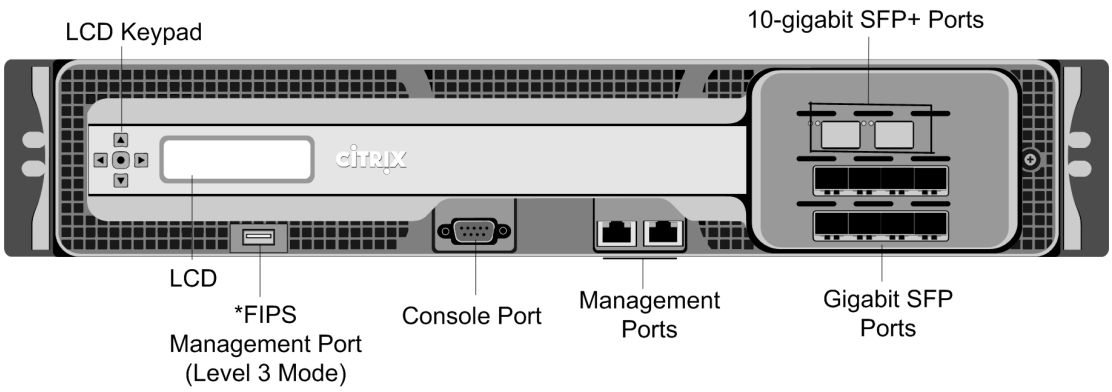


Figure 3. Citrix NetScaler MPX 9700/10500/12500/15500 FIPS, front panel

*The FIPS Management Port (Level 3 Mode) is reserved for a future release.

Caution: Do not insert a USB device into the FIPS Management Port. This will cause the FIPS card to fail.

Depending on the model, the appliance has the following ports:

- FIPS Management Port (reserved for a future release).
- RS232 serial Console Port.
- Two 10/100/1000Base-T copper Ethernet Management Ports (RJ45), numbered 0/1 and 0/2 from left to right. These ports are used to connect directly to the appliance for system administration functions.
- Network Ports
  - MPX 9700/10500/12500/15500. Eight 1-gigabit copper or fiber SFP ports numbered 1/1, 1/2, 1/3, and 1/4 on the first row from left to right, and 1/5, 1/6, 1/7, and 1/8 on the second row from left to right. Eight 10/100/1000BASE-T copper Ethernet Ports (RJ45) numbered 1/9, 1/10, 1/11, and 1/12 on the third row from left to right, and 1/13, 1/14, 1/15, and 1/16 on the fourth row from left to right.
  - MPX 9700/10500/12500/15500 10G and MPX 9700/10500/12500/15000 FIPS. Two 10-gigabit SFP+ Ports numbered 10/1 and 10/2 on the top row, eight 1-gigabit copper or fiber SFP Ports numbered 1/1, 1/2, 1/3, and 1/4 on the middle row from left to right, and 1/5, 1/6, 1/7, and 1/8 on the bottom row from left to right.

Important: The 10-gigabit ports on this appliance are labeled 10/1 and 10/2.

The following figure shows the back panel of the MPX 9700/10500/12500/15500 appliances, including the 10G model and FIPS model.

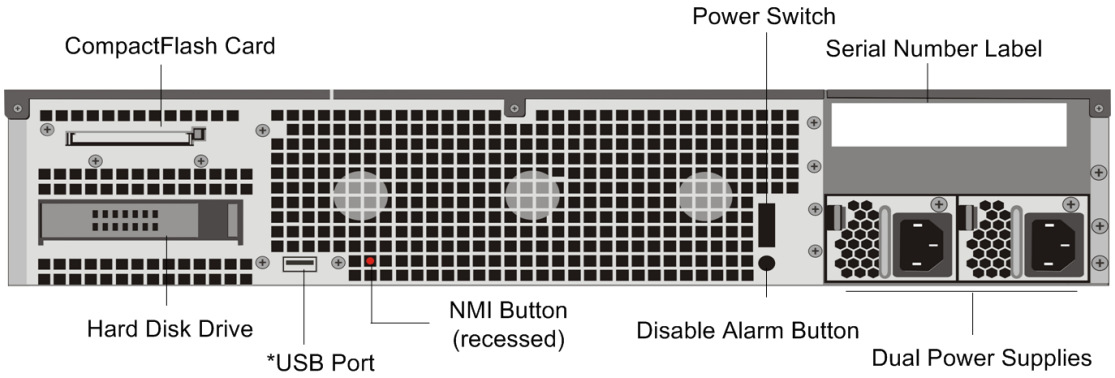


Figure 4. Citrix NetScaler MPX 9700/10500/12500/15500, MPX 9700/10500/12500/15500 FIPS, and MPX 9700/10500/12500/15500 10G, back panel

*The USB Port is reserved for a future release.

The following components are visible on the back panel of the MPX 9700/10500/12500/15500, including the 10G model and FIPS model:

- Four GB removable CompactFlash Card that is used to store the operating system.
- Power Switch, which turns off power to the appliance, just as if you were to unplug the power supply. Press the switch for five seconds to turn off the power.
- Removable Hard Disk Drive that is used to store user data.
- USB Port (reserved for a future release).
- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the NetScaler. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.
- Disable Alarm Button. This button is functional only when the appliance has two power supplies.

  Press this button to stop the power alarm from sounding when you have plugged the appliance into only one power outlet or when one power supply is malfunctioning and you want to continue operating the appliance until it is repaired.
- Dual Power Supplies, each rated at 450 watts, 110-220 volts.

Citrix NetScaler MPX 11500, MPX 13500, MPX 14500, MPX 16500, MPX 18500, and MPX 20500

The Citrix NetScaler models MPX 11500/13500/14500/16500/18500/20500 are 2U appliances. Each model has two 6-core processors for a total of 12 physical cores (24 cores with hyper-threading), and 48 gigabytes (GB) of memory.

The following figure shows the front panel of the MPX 11500/13500/14500/16500/18500/20500.

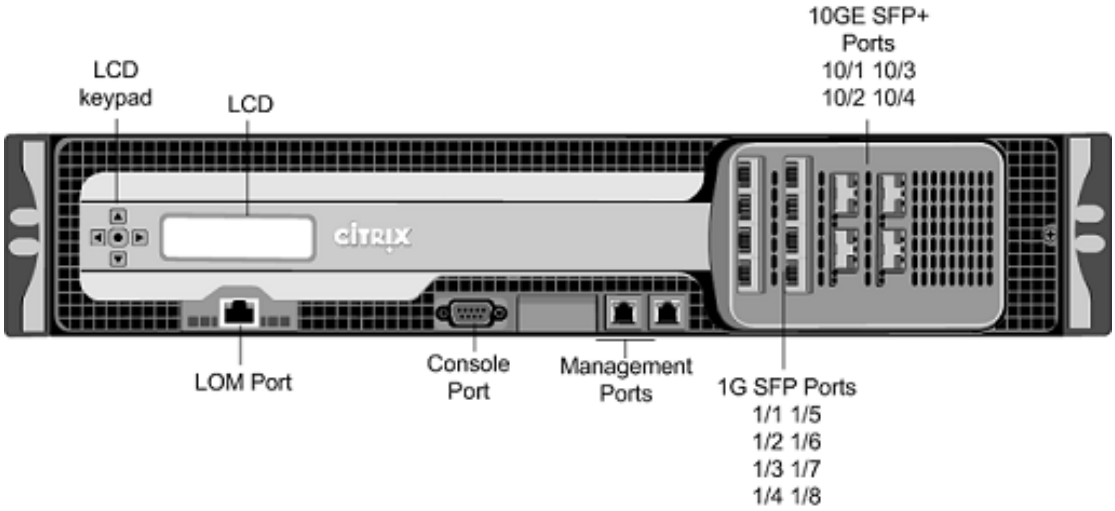


Figure 1. Citrix NetScaler MPX 11500/13500/14500/16500/18500/20500, front panel

The MPX 11500/13500/14500/16500/18500/20500 appliances have the following ports:

- 10/100Base-T copper Ethernet Port (RJ45), also called LOM Port. You can use this port to remotely monitor and manage the appliance independently of the operating system.

  Note: The LEDs on the LOM port are not operational by design.

- RS232 serial Console Port.

- Two 10/100/1000Base-T copper Ethernet Management Ports (RJ45), numbered 0/1 and 0/2 from left to right. These ports are used to connect directly to the appliance for system administration functions.

- Eight 1-gigabit SFP ports numbered 1/1, 1/2, 1/3, 1/4 from top to bottom in the first column, and 1/5, 1/6, 1/7, and 1/8 from top to bottom in the second column.

- Four 10-gigabit SFP+ Ports numbered 10/1 and 10/2 from top to bottom in the first column, and 10/3 and 10/4 from top to bottom in the second column.

The following figure shows the back panel of the MPX 11500/13500/14500/16500/18500/20500.

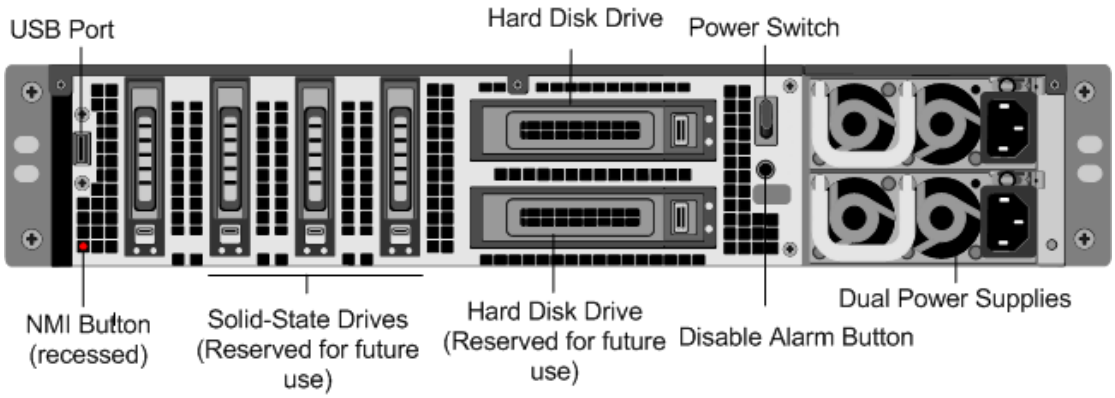


Figure 2. Citrix NetScaler MPX 11500/13500/14500/16500/18500/20500, back panel

The following components are visible on the back panel of the MPX 11500/13500/14500/16500/18500/20500:

- 160 GB removable Solid-State Drive that is used to store the operating system.
- USB port (reserved for a future release).
- Power Switch, which turns off power to the appliance, just as if you were to unplug the power supply. Press the switch for five seconds to turn off the power.
- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the NetScaler. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.
- Two removable Hard Disk Drives that are used to store user data.
- Disable Alarm Button. This button is functional only when the appliance has two power supplies.

  Press this button to stop the power alarm from sounding when you have plugged the appliance into only one power outlet or when one power supply is malfunctioning and you want to continue operating the appliance until it is repaired.
- Dual Power Supplies, each rated at 650 watts, 110-220 volts.

Citrix NetScaler MPX 15000

The Citrix NetScaler MPX 15000 appliance is a 2U appliance, with 2 dual-core processors, and 16 GB of memory. The MPX 15000 is a high-capacity, fault-tolerant hardware platform intended for heavy use in enterprise and service provider environments. The following figure shows the front panel of the MPX 15000.

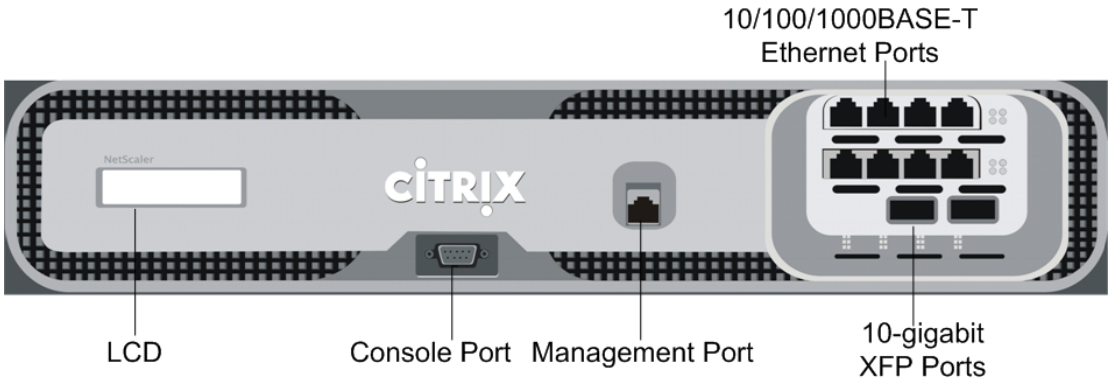


Figure 1. Citrix NetScaler MPX 15000, front panel

The appliance has the following ports:

- RS232 serial Console Port.
- 10/100/1000BASE-T copper Ethernet Management Port, numbered 0/1.
- Two XFP (10-Gigabit Small Form Factor Pluggable) fiberoptic ports, numbered from left to right 1/1 and 1/2.
- Eight 10/100/1000BASE-T copper Ethernet ports, numbered from upper left to bottom right 1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9, and 1/10.

When facing the bezel, the upper LEDs to the left of each port represent connectivity. They are lit and amber in color when active. The lower LEDs represent throughput. They are lit and green when active.

Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number and will always be either 0 or 1. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

The following figure shows the back panel of the MPX 15000.

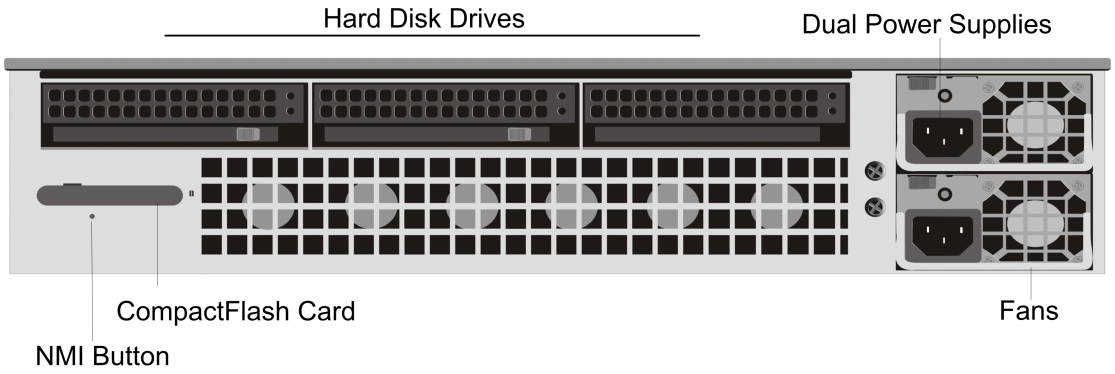


Figure 2. Citrix NetScaler MPX 15000, back panel

The following components are visible on the back panel of the MPX 15000:

- Removable Hard Disk Drive that is used to store user data.
- Dual Power supplies, each rated at 500 watts, 110-220 volts.

  You plug separate power cords into the power supplies and connect them to separate wall sockets. The MPX 15000 functions properly with a single power supply; the extra power supply serves as a backup.
- Non-maskable interrupt (NMI) button, which signals the MPX 15000 to perform an orderly shutdown after saving all files. You must use a pen, pencil, or other pointed object to press this button, which is located inside a small hole to prevent it being pressed accidentally.
- Removable CompactFlash Card that is used to store the operating system.

Citrix NetScaler MPX 17000

The Citrix NetScaler MPX 17000 appliance is a 2U appliance, with 2 quad-core processors, and 32 GB of memory. The MPX 17000 is a high-capacity, fault-tolerant hardware platform intended for any high traffic, intensive processing data center environment. There are two MPX 17000 models: the four network-port model and the ten network-port model. The following figure shows the front panel of the MPX 17000, four network-port model.

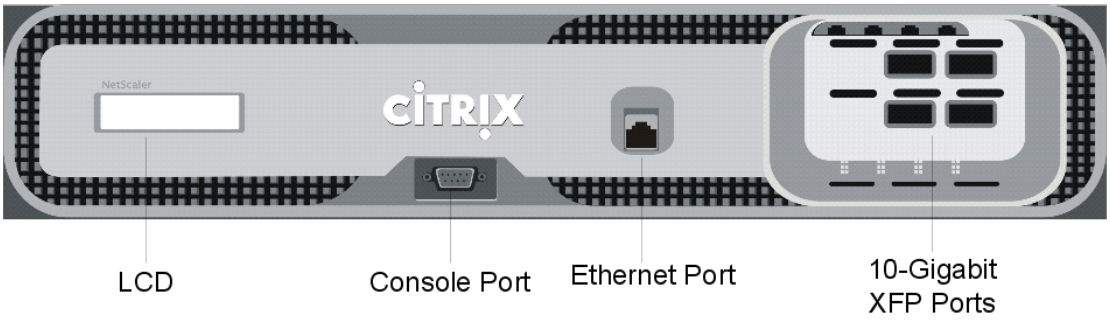


Figure 1. Citrix NetScaler MPX 17000 four network-port model, front panel

Depending on the model, the appliance has the following ports:

- RS232 serial Console Port.
- 10/100/1000BASE-T copper Ethernet Management Port, numbered 0/1.
- Network Ports
    - MPX 17000 four network-port model. Four XFP (10-Gigabit Small Form Factor Pluggable) ports, numbered from upper left to bottom right 1/1, 1/2, 1/3, and 1/4.
    - MPX 17000 ten network-port model. Two XFP ports, numbered from left to right 1/1 and 1/2 and eight 10/100/1000BASE-T Ethernet ports, numbered from upper left to bottom right 1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9 and 1/10.

    Note: The network port numbers on all appliances consist of two numbers separated by a forward slash. The first number is the port adapter slot number and will always be either 0 or 1. The second number is the interface port number. Ports on appliances are numbered sequentially starting with 1.

When facing the bezel, the upper LEDs to the left of each port represent connectivity. They are lit and amber in color when active. The lower LEDs represent throughput. They are lit and green when active.

The following figure shows the back panel of the MPX 17000.

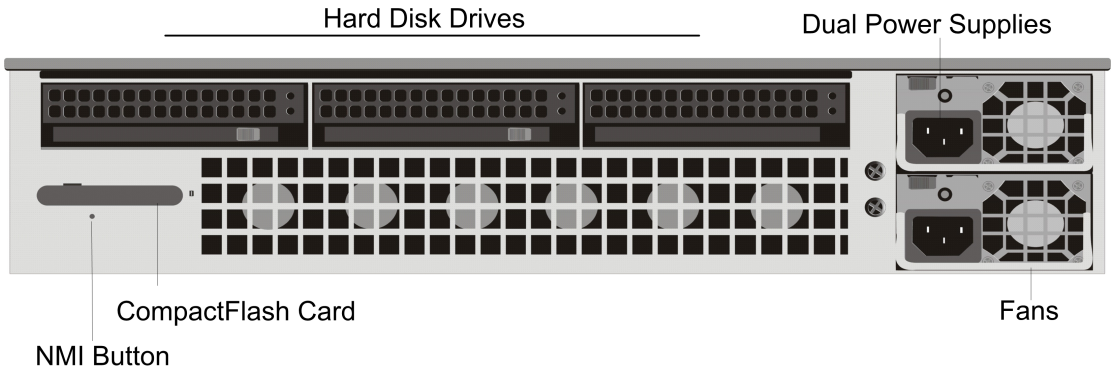


Figure 2. Citrix NetScaler MPX 17000, back panel

The following components are visible on the back of the MPX 17000:

- Removable Hard Disk Drive that is used to store user data.
- Dual Power supplies, each rated at 500 watts, 110-220 volts.

  You plug separate power cords into the power supplies and connect them to separate wall sockets. The MPX 17000 functions properly with a single power supply; the extra power supply serves as a backup.
- Non-maskable interrupt (NMI) button, which signals the MPX 17000 to perform an orderly shutdown after saving all files. You must use a pen, pencil, or other pointed object to press this button, which is located inside a small hole to prevent it being pressed accidentally.
- Removable CompactFlash Card that is used to store the operating system.

Citrix NetScaler MPX 17500, MPX 19500, and MPX 21500

The Citrix NetScaler models MPX 17500/19500/21500 are 2U appliances. Each model has two 6-core processors and 48 gigabytes (GB) of memory.

The following figure shows the front panel of the MPX 17500/19500/21500.

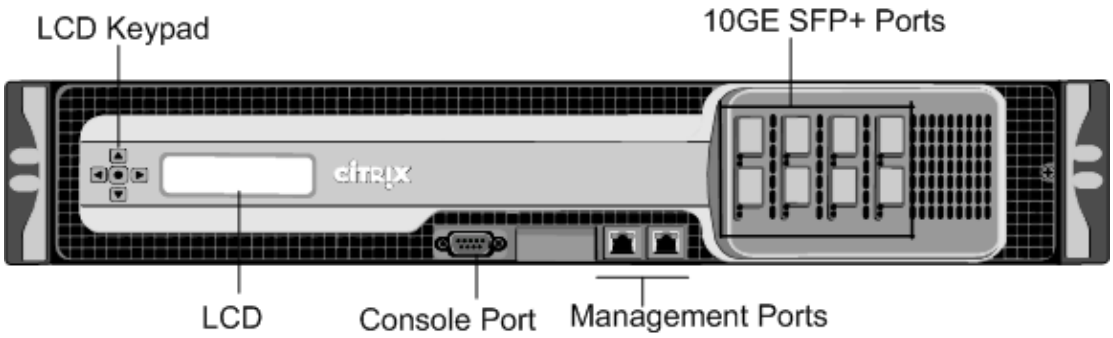


Figure 1. Citrix NetScaler MPX 17500/19500/21500, front panel

The MPX 17500/19500/21500 appliances have the following ports:

- RS232 serial Console Port.
- Two 10/100/1000Base-T copper Ethernet Management Ports (RJ45), numbered 0/1 and 0/2 from left to right. These ports are used to connect directly to the appliance for system administration functions.
- Eight 10-gigabit SFP+ Ports numbered 10/1, 10/2, 10/3, and 10/4 on the top row from left to right, and 10/5, 10/6, 10/7, and 10/8 on the bottom row from left to right.

The following figure shows the back panel of the MPX 17500/19500/21500.

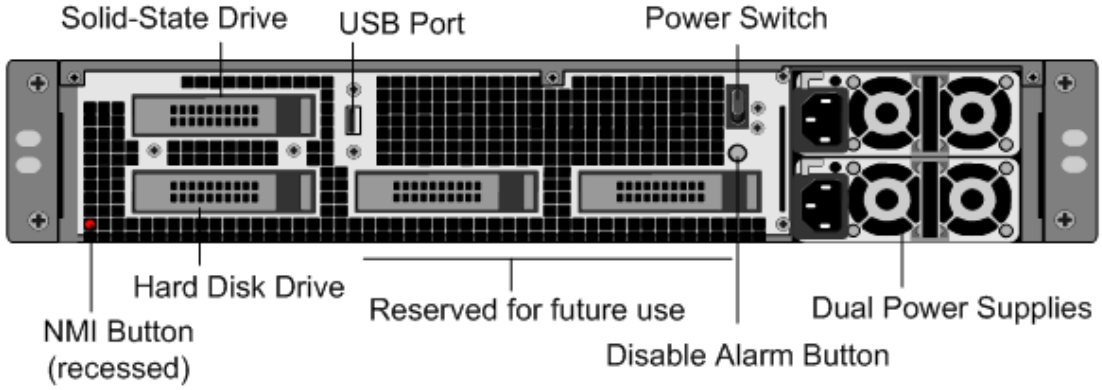


Figure 2. Citrix NetScaler MPX 17500/19500/21500, back panel

The following components are visible on the back panel of the MPX 17500/19500/21500:

- 160 GB removable Solid-State Drive that is used to store the operating system.
- USB port (reserved for a future release).
- Power Switch, which turns off power to the appliance, just as if you were to unplug the power supply. Press the switch for five seconds to turn off the power.
- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the NetScaler. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.
- Removable Hard Disk Drive that is used to store user data.
- Disable Alarm Button. This button is functional only when the appliance has two power supplies.

  Press this button to stop the power alarm from sounding when you have plugged the appliance into only one power outlet or when one power supply is malfunctioning and you want to continue operating the appliance until it is repaired.
- Dual Power Supplies, each rated at 650 watts, 110-220 volts.

Citrix NetScaler MPX 17550, MPX 19550, MPX 20550, and MPX 21550

The Citrix NetScaler models MPX 17550, MPX 19550, MPX 20550, and MPX 21550 are 2U appliances. Each model has two 6-core processors for a total of 12 physical cores (24 cores with hyper-threading), and 96 gigabytes (GB) of memory.

The following figure shows the front panel of the MPX 17550/19550/20550/21550 appliance.

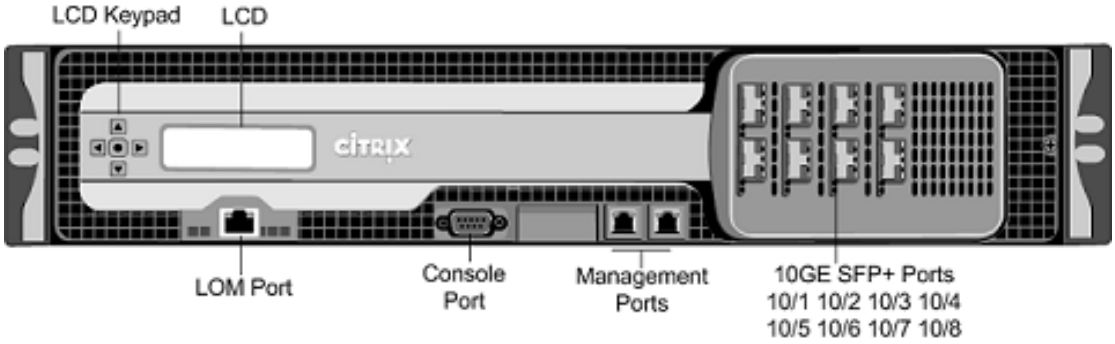


Figure 1. Citrix NetScaler MPX 17550/19550/20550/21550 appliance, front panel

The MPX 17550/19550/20550/21550 appliance has the following ports:

- 10/100Base-T copper Ethernet Port (RJ45), also called LOM Port. You can use this port to remotely monitor and manage the appliance independently of the operating system.

  Note: The LEDs on the LOM port are not operational by design.

- RS232 serial Console Port.
- Two 10/100/1000Base-T copper Ethernet Management Ports (RJ45), numbered 0/1 and 0/2 from left to right. These ports are used to connect directly to the appliance for system administration functions.
- Eight 10-gigabit SFP+ Ports numbered 10/1, 10/2, 10/3, and 10/4 on the top row from left to right, and 10/5, 10/6, 10/7, and 10/8 on the bottom row from left to right.

The following figure shows the back panel of the MPX 17550/19550/20550/21550 appliance.

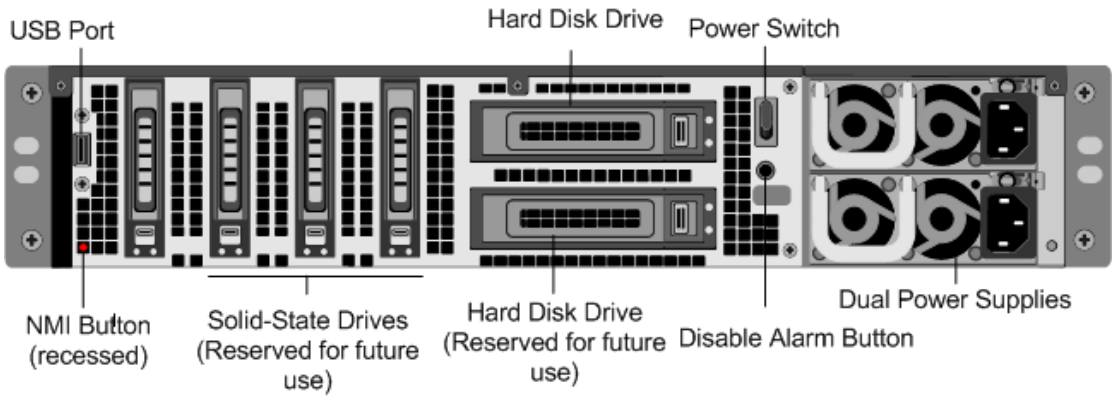


Figure 2. Citrix NetScaler MPX 17550/19550/20550/21550 appliance, back panel

The following components are visible on the back panel of the MPX 17550/19550/20550/21550 appliance:

- 160 GB removable Solid-State Drive that is used to store the operating system.
- USB port (reserved for a future release).
- Power Switch, which turns off power to the appliance, just as if you were to unplug the power supply. Press the switch for five seconds to turn off the power.
- Non-maskable interrupt (NMI) Button that is used at the request of Technical Support and produces a core dump on the NetScaler. You must use a pen, pencil, or other pointed object to press this red button, which is recessed to prevent unintentional activation.
- Two removable Hard Disk Drives that are used to store user data.
- Disable Alarm Button. This button is functional only when the appliance has two power supplies.

  Press this button to stop the power alarm from sounding when you have plugged the appliance into only one power outlet or when one power supply is malfunctioning and you want to continue operating the appliance until it is repaired.
- Dual Power Supplies, each rated at 850 watts, 110-220 volts.

Summary of Hardware Specifications

The following tables summarize the specifications of the hardware platforms.

Table 1. Standard Platform Summary

	7000	9010	10010	12000
Processor	1 single-core	1 single-core	1 single-core	2 single-cores
Memory	1 GB	2 GB	4 GB	4 GB
Number of Power Supplies	1	2	2	2
Power Supply input voltage & frequency	100-240 VAC 47-63 Hz	100-240 VAC 47-63 Hz	100-240 VAC 47-63 Hz	100-240 VAC 47-63 Hz
Maximum Power Consumption	250 W	500 W	500 W	500 W
Weight	28 lbs	52 lbs	52 lbs	52 lbs
Height	1U	2U	2U	2U
Width	EIA 310-D EIA	EIA 310-D	EIA 310-D	EIA 310-D
Depth	24 in or 61 cm	24 in or 61 cm	24 in or 61 cm	24 in or 61 cm
Operating Temperature (degree celsius)	0-40	0-40	0-40	0-40
Humidity range (non-condensing)	5%-95%	5%-95%	5%-95%	5%-95%
Safety Certifi-cations	TUV	TUV	TUV	TUV
EMC & Susceptibility	FCC Class A, CB, FCC, CE, VCCI, C-Tick, NOM, SASO/CITC	FCC Class A, CB, FCC, CE, VCCI, C-Tick, NOM	FCC Class A, CB, FCC, CE, VCCI, C-Tick, NOM, SASO/CITC	FCC Class A, CB, FCC, CE, VCCI, C-Tick, NOM, SASO/CITC
Compliance	RoHS, WEEE	RoHS, WEEE	RoHS, WEEE	RoHS, WEEE

Table 2. MPX Platform Summary

	MPX 5500	MPX 7500/MPX 9500	MPX 15000	MPX 17000
Processors	1 dual-core	1 dual-core	2 quad-core	2 quad-core
Memory	4 GB	8 GB	16 GB	32 GB

Number of Power Supplies	1	1 with second optional	2	2
AC Power Supply input voltage, frequency, & current	100-240 VAC 50-60 Hz 3-1.5 A	100-240 VAC 50-60 Hz 3-1.5 A	100-240 VAC 47-63 Hz	100-240 VAC 47-63 Hz
Maximum Power Consumption	260 W	260 W	700 W	700 W
Heat Dissipation	887 BTU per hour	887 BTU per hour		
Weight	22 lbs	23 lbs with one power supply	52 lbs	52 lbs
Height	1U	1U	2U	2U
Width	EIA 310-D for 19-inch racks	EIA 310-D for 19-inch racks	EIA 310-D for 19-inch racks	EIA 310-D for 19-inch racks
Depth	21.75 in or 55 cm	21.75 in or 55 cm	18.5 in or 47 cm	18.5 in or 47 cm
Operating Temperature (degree Celsius)	0-40	0-40	0-35	0-35
Humidity range (non-condensing)	5%-95%	5%-95%	5%-95%	5%-95%
Safety Certifi-cations	CSA	CSA	UL & TUV-C	UL & TUV-C
EMC & Susceptibility	FCC (Part 15 Class A), CE, C-Tick, CCC, KCC, NOM, PCT, VCCI, SASO, SABS	FCC (Part 15 Class A), CE, C-Tick, CCC, KCC, NOM, PCT, VCCI, SASO, SABS	FCC (Part 15 Class A), DoC, CE, VCCI, CNS, AN/NES	FCC (Part 15 Class A), DoC, CE, VCCI, CNS, AN/NES
Compliance	RoHS, WEEE	RoHS, WEEE	RoHS, WEEE	RoHS, WEEE

Table 3. MPX Platform Summary (contd.)

	MPX 9700/MPX 10500/MPX 12500/MPX 15500	MPX 11500/MPX 13500/MPX 14500/MPX 16500/MPX 18500/ MPX 20500	MPX 17500/MPX 19500/MPX 21500	MPX 17550/MPX 19550/MPX 20550/MPX 21550
Processors	2 quad-core	2 six-core	2 six-core	2 six-core
Memory	16 GB	48 GB	48 GB	96 GB

Number of Power Supplies	2	2	2	2
AC Power Supply input voltage, frequency, & current	100-240 VAC 50-60 Hz 4.5-2.5 A	100-240 VAC 50-60 Hz 6.5-3.5 A	100-240 VAC 50-60 Hz 6.5-3.5 A	100-240 VAC 50-60 Hz 6.5-3.5 A
Maximum Power Consumption	450 W	650 W	650 W	850 W
Heat Dissipation	1550 BTU per hour	2200 BTU per hour	2200 BTU per hour	2900 BTU per hour
Weight	31 lbs	46 lbs	40 lbs	40 lbs
Height	2U	2U	2U	2U
Width	EIA 310-D for 19-inch racks	EIA 310-D for 19-inch racks	EIA 310-D for 19-inch racks	EIA 310-D for 19-inch racks
Depth	24.5 in or 62 cm	28 in or 71.68 cm	24.75 in or 62.865 cm	24.75 in or 62.865 cm
Operating Temperature (degree Celsius)	0-40	0-40	0-40	0-40
Humidity range (non-condensing)	5%-95%	5%-95%	5%-95%	5%-95%
Safety Certifi-cations	CSA	CSA	TUV	TUV
EMC & Susceptibility	FCC (Part 15 Class A), CE, C-Tick, KCC, NOM, PCT, VCCI, SASO, SABS	FCC (Part 15 Class A), CE, C-Tick, VCCI, CCC, KC, NOM, GOST, SABS, SASO	FCC (Part 15 Class A), CE, C-Tick, VCCI-A	FCC (Part 15 Class A), CE, C-Tick, VCCI-A
Compliance	RoHS, WEEE	RoHS, SVHC, WEEE	RoHS, WEEE	RoHS, WEEE

Preparing for Installation

Before you install your new NetScaler appliance, carefully unpack your appliance and make sure that all parts were delivered. Once you are satisfied that your appliance has been delivered to your expectations, verify that the location where the appliance will be installed meets temperature and power requirements and that the server cabinet or floor-to-ceiling cabinet is securely bolted to the floor and has sufficient airflow.

Only trained and qualified personnel should install, maintain, or replace the appliance and efforts should be taken to ensure that all cautions and warnings are followed.

Unpacking the NetScaler Appliance

The hardware accessories for your particular appliance, such as cables, adapters, and rail kit, will vary depending on the hardware platform you ordered. Unpack the box that contains your new appliance on a sturdy table with plenty of space and inspect the contents.

Use the following list to verify that you received everything that should have been included in the box.

- The appliance you ordered For a description and illustration of your particular model, see Hardware Platforms.
- One RJ-45 to DB-9 adapter
- One 6 ft RJ-45/DB-9 cable
- The following list specifies the number of power cables included for each appliance model:
  - One power cable for the 7000, MPX 5500, and MPX 7500/9500 appliances
  - Two power cables for the 9010, 10010, 12000, MPX 15000, MPX 17000, MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances

    Note: Make sure that a power outlet is available for each cable.
- One mounting rail kit with all the models

In addition to the items included in the box with your new appliance, you will need the following items to complete the installation and initial configuration process.

- Ethernet cables for each additional Ethernet port that you will connect to your network
- One available Ethernet port on your network switch or hub for each NetScaler Ethernet port you want to connect to your network

  Note: Transceiver modules are sold separately. Please contact your Citrix sales representative to order transceiver modules for your appliance. Only transceivers supplied by Citrix are supported on the appliance.
- A computer to serve as a management workstation

Preparing the Site and Rack

There are specific site and rack requirements for the NetScaler appliance. You must make sure that adequate environmental control and power density are available. Racks must be bolted to the ground, have sufficient airflow, and have adequate power and network connections. Preparing the site and rack are important steps in the installation process and will help ensure a smooth installation.

Site Requirements

The appliance should be installed in a server room or server cabinet with the following features:

Environment control

An air conditioner, preferably a dedicated computer room air conditioner (CRAC), capable of maintaining the cabinet or server room at a temperature of no more than 21 degrees C/70 degrees F at altitudes up to 2100 m/7000 ft, or 15 degrees C/60 degrees F at higher altitudes, a humidity level no greater than 45 percent, and a dust-free environment.

Power density

Wiring capable of handling at least 4,000 watts per rack unit in addition to power needs for the CRAC.

Rack Requirements

The rack on which you install your appliance should meet the following criteria:

Rack characteristics

Racks should be either integrated into a purpose-designed server cabinet or be the floor-to-ceiling type, bolted down at both top and bottom to ensure stability. If you have a cabinet, it should be installed perpendicular to a load-bearing wall for stability and sufficient airflow. If you have a server room, your racks should be installed in rows spaced at least 1 meter/3 feet apart for sufficient airflow. Your rack must allow your IT personnel unfettered access to the front and back of each server and to all power and network connections.

Power connections

At minimum, two standard power outlets per unit.

Network connections

At minimum, four Ethernet connections per rack unit.

Space requirements

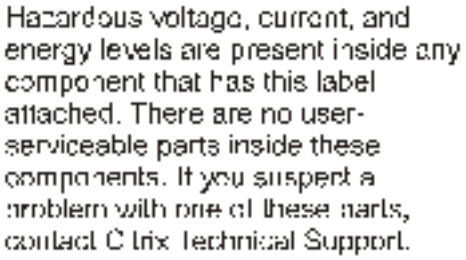
One empty rack unit for the Citrix NetScaler 7000, MPX 5500, and MPX 7500/MPX 9500, and two consecutive empty rack units for all oter appliance models.

Cautions and Warnings

Electrical Safety Precautions

Basic electrical safety precautions should be followed to protect yourself from harm and the appliance from damage.

- Be aware of the location of the emergency power off (EPO) switch so that if an electrical accident occurs, you can quickly remove power to the appliance.
- Remove all jewelry and other metal objects that might come into contact with power sources or wires before installing or repairing the appliance. When you touch both a live power source or wire and ground, any metal objects can heat up rapidly, and may cause burns, set clothing on fire, or fuse the metal object to an exposed terminal.
- Use a regulating uninterruptible power supply (UPS) to protect the appliance from power surges and voltage spikes, and to keep the appliance operating in case of power failure.
- Never stack the appliance on top of any other server or electronic equipment.
- All appliances are designed to be installed on power systems that use TN earthing. Do not install your device on a power system that uses either TT or IT earthing.
- Make sure that the appliance has a direct physical connection to the earth during normal use. When installing or repairing an appliance, always make sure that the ground circuit is connected first and disconnected last.
- Make sure that a fuse or circuit breaker no larger than 120 VAC, 15 A U.S. (240 VAC, 16 A international) is used on all current-carrying conductors on the power system to which your appliances are connected.
- Do not work alone when working with high voltage components.
- Always disconnect the appliance from power before removing or installing any component. When disconnecting power, you should first shut down the appliance and then unplug the power cords of all the power supply units connected to the appliance. As long as the power cord is plugged in, line voltages may be present in the power supply, even when the power switch is OFF.
- Do not use mats designed to decrease static electrical discharge as protection from electrical shock. Instead, use rubber mats that have been specifically designed as electrical insulators.
- Make sure that the power source can handle the appliance's maximum power consumption rating with no danger of an overload. Always unplug any appliance before performing repairs or upgrades.
- Do not overload the wiring in your server cabinet or on your server room rack.

- During thunderstorms, or anticipated thunderstorms, avoid performing any hardware repairs or upgrades until the danger of lightning has passed.
- When you dispose of an old appliance or any components, follow any local and national laws on disposal of electronic waste.
- To prevent possible explosions, replace expired batteries with the same model or a manufacturer-recommended substitute and follow the manufacturer's instructions on battery replacement.
- 

Never remove a power supply cover or any sealed part that has the following label:

Appliance Precautions

- Determine the placement of each component in the rack before you install the rail.
- Install the heaviest appliance first at the bottom of the rack, and then work upward. Distribute the load on the rack evenly. An unbalanced rack is hazardous.
- Allow the power supply units and hard drives to cool before touching them.
- Install the equipment near a socket outlet for easy access.
- Mount equipment into a rack with sufficient airflow for safe operation.
- For a closed or multi-unit rack assembly, the ambient operating temperature of the rack environment may be greater than the ambient temperature of the room. Therefore, consider the lowest and highest operating temperatures of the equipment when making a decision about where to install the appliance in the rack.

Rack Precautions

- Make sure that the leveling jacks on the bottom of the rack are fully extended to the floor, with the full weight of the rack resting on them.
- For a single-rack installation, attach a stabilizer to the rack.
- For a multiple-rack installation, couple (attach) the racks together.
- Always make sure that the rack is stable before extending a component from the rack.
- Extend only one component at a time. Extending two or more simultaneously may cause the rack to become unstable.

- The handles on the left and right of the front panel of the appliance should only be used for extending the appliance out of the rack. These handles should not be used for mounting the appliance on the rack. Rack-rail hardware described later should be used instead.

Installing the Hardware

After you have determined that the location where you will install your appliance meets the environmental standards and the server rack is in place according to the instructions, you are ready to install the hardware. After you mount the appliance, you are ready to connect it to the network, to a power source, and to the console terminal that you will use for initial configuration. To complete the installation, you turn on the appliance. Be sure to observe the cautions and warnings listed with the installation instructions.

Rack Mounting the Appliance

Most appliances can be installed in standard server racks that conform to EIA-310-D specification. The appliances ship with a set of rails, which you must install before you mount the appliance. The only tools that you need for installing an appliance are a Phillips screwdriver and a flathead screwdriver.

Caution: If you are installing the appliance as the only unit in the rack, mount it at the bottom. If the rack contains other units, make sure that the heaviest unit is at the bottom. If the rack has stabilizing devices available, install them before mounting the appliance.

The following table lists the different hardware platforms and the rack units required for each platform.

Table 1. *Height Requirements For Each Platform*

Platform	Number of rack units
7000	One rack unit
9010, 10010, 12000	Two rack units
MPX 5500	One rack unit
MPX 7500/9500	One rack unit
MPX 9700/10500/12500/15500	Two rack units
MPX 15000, MPX 17000	Two rack units
MPX 11500/13500/14500/16500/18500/20500	Two rack units
MPX 17500/19500/21500	Two rack units
MPX 17550/19550/20550/21550	Two rack units

Each appliance ships with a mounting rail kit that contains two rail assemblies, one for the left side and the other for the right side of the appliance, and screws to attach the rails. An assembly consists of an inner rail and a rack rail.

Note: The same rail kit is used for both square-hole and round-hole racks. See Figure 4 for specific instructions for threaded, round-hole racks.

To mount the appliance, you must first install the rails and then install the appliance in the rack.

Perform the following tasks to mount the appliance:

- Remove the inner rails from the rail assembly.
- Attach the inner rails to the appliance.
- Install the rack rails on the rack.

- Install the appliance in the rack.

To remove the inner rails from the rail assembly

1. Place the rail assembly on a flat surface.
2. Slide out the inner rail toward the front of the assembly.
3. Depress the latch until the inner rail comes all the way out of the rail assembly.
4. Repeat steps 1 through 3 to remove the second inner rail.

To attach the inner rails to the appliance

1. Position the right inner rail behind the handle on the right side of the appliance.
2. Align the holes on the rail with the corresponding holes on the side of the appliance.
3. Attach the rail to the appliance with the provided screws: 4 per side for a 1U appliance and 5 per side for a 2U appliance, as shown in the following figure.

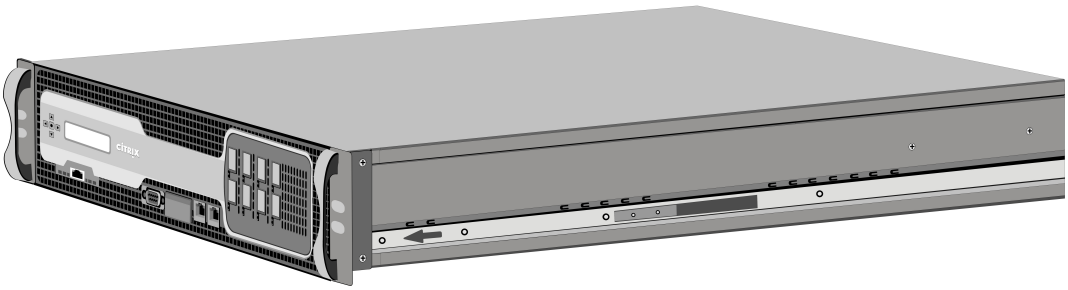


Figure 1. Attaching inner rails

4. Repeat steps 1 through 3 to install the left inner rail on the other side of the appliance.

To install the rack rails on the rack

1. If you have a round-hole, threaded rack, skip to step 3.

2. Install square nut retainers into the front post and back post of the rack as shown in the following figures. Before inserting a screw, be sure to align the square nut with the correct hole for your 1U or 2U appliance. The three holes are not evenly spaced.

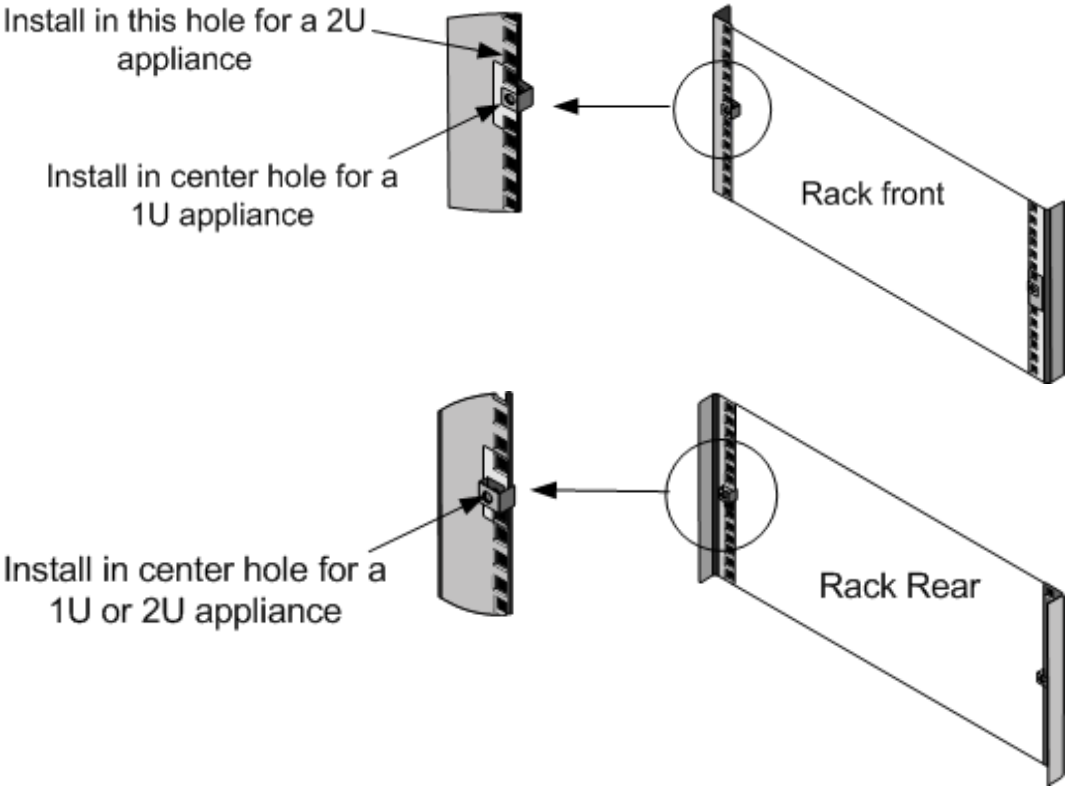


Figure 2. Installing Retainers into the Front Rack Posts Figure 3. Installing Retainers into the Rear Rack Posts

3. Install the adjustable rail assembly into the rack as shown in the following figures. Use a screw to lock the rear rail flange into the rack. With the screw securing the rail in place, you can optionally remove the latching spring.

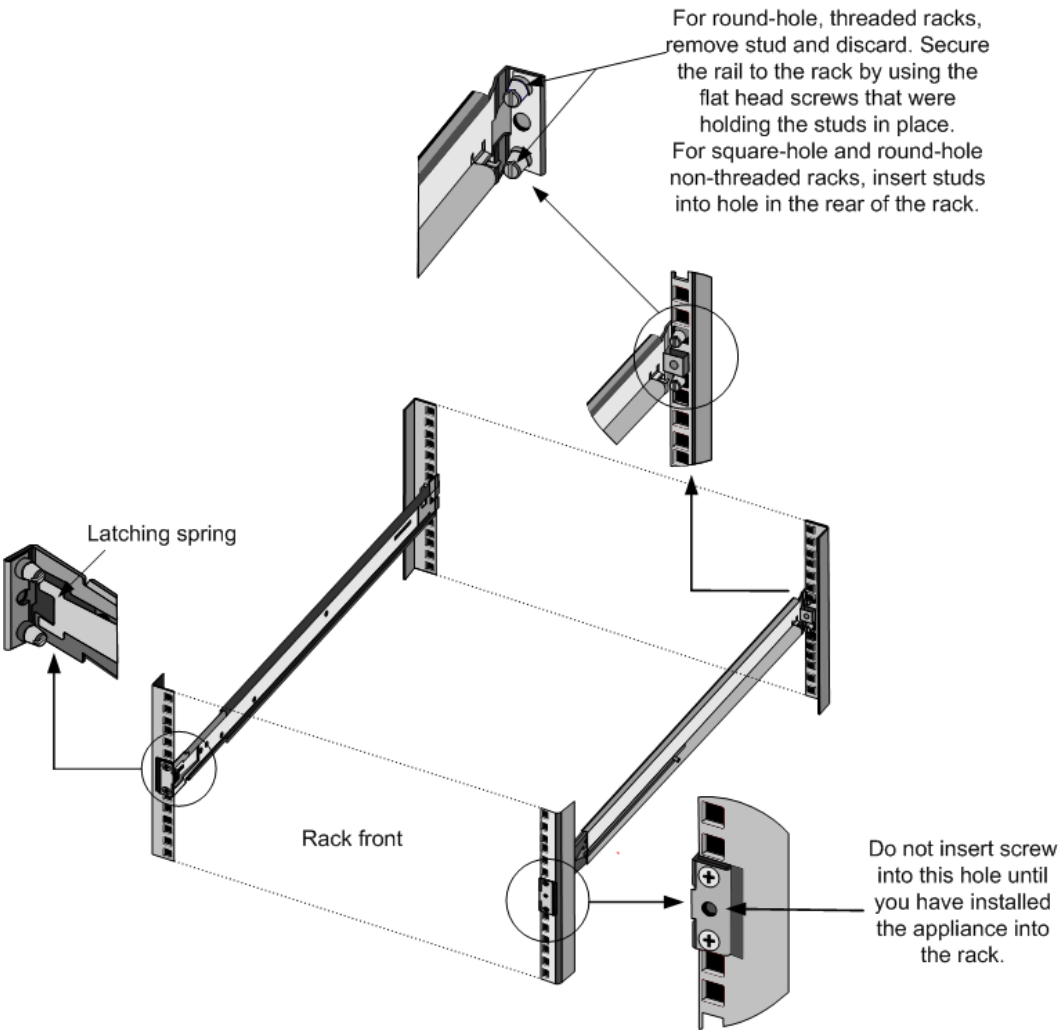


Figure 4. Installing the Rail Assembly to the Rack

To install the appliance in the rack

1. Align the inner rails, attached to the appliance, with the rack rails.
2. Slide the appliance into the rack rails, keeping the pressure even on both sides.
3. Verify that the appliance is locked in place by pulling it all the way out from the rack.

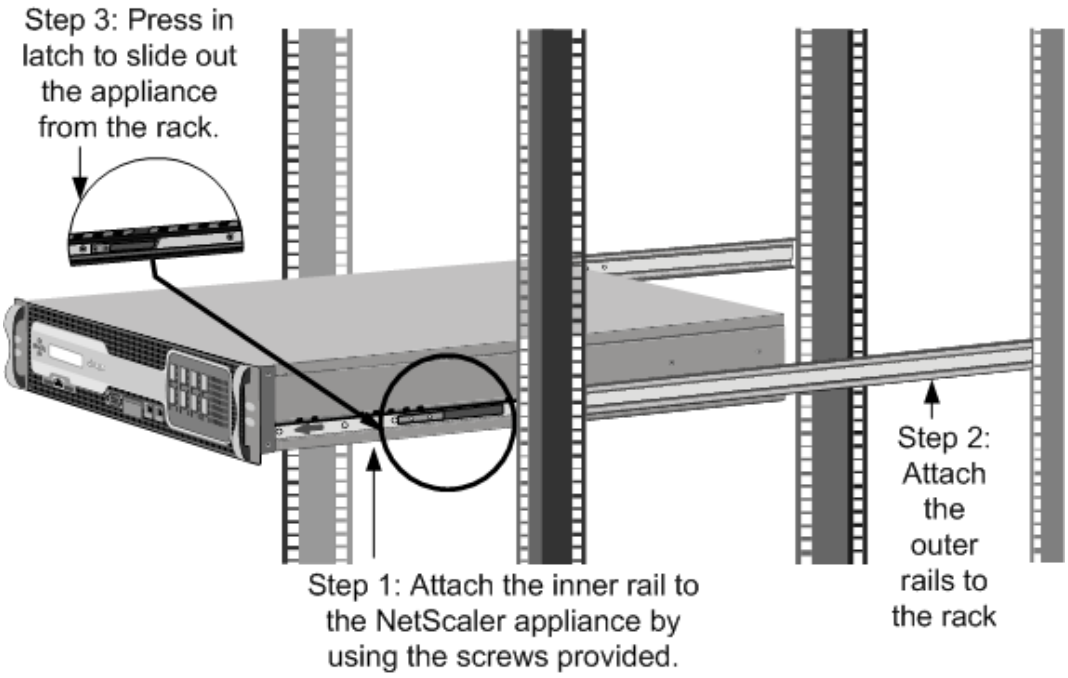


Figure 5. Rack Mounting the Appliance

Installing and Removing SFP Transceivers

Note: This section applies to the 9010, 10010, 12000, MPX 9700/10500/12500/15500, and MPX 11500/13500/14500/16500/18500/20500 appliances.

A Small Form Factor Pluggable (SFP) is a compact transceiver that can operate at speeds of up to 1 gigabit per second and is available in both copper and fiber types. Inserting an SFP copper transceiver converts the SFP port to a 1000BASE-T port. Inserting an SFP fiber transceiver converts the SFP port to a 1000BASE-X port. Auto-negotiation is enabled by default on the SFP port into which you insert your SFP transceiver. As soon as a link between the port and the network is established, the speed and mode are matched on both ends of the cable.

Caution: Only SFP transceivers provided by Citrix Systems are supported on NetScaler appliances. Attempting to install third-party SFP transceivers on your NetScaler appliance voids the warranty.

Insert SFP transceivers into the SFP ports on the front panel of the appliance. Frequent installation and removal of transceivers shortens their life span. Follow the removal procedure carefully to avoid damaging the SFP transceiver or the appliance.

Caution: Do not install the transceivers with the cables attached. Doing so can damage the cable, the connector, or the optical interface of the transceiver.

To install an SFP transceiver

1. Remove the SFP transceiver carefully from its box. DANGERDo not look directly into fiberoptic transceivers or cables. They emit laser beams that can damage your eyes.

2. Align the SFP transceiver to the front of the SFP transceiver port on the front panel of the appliance, as shown in the following figure.

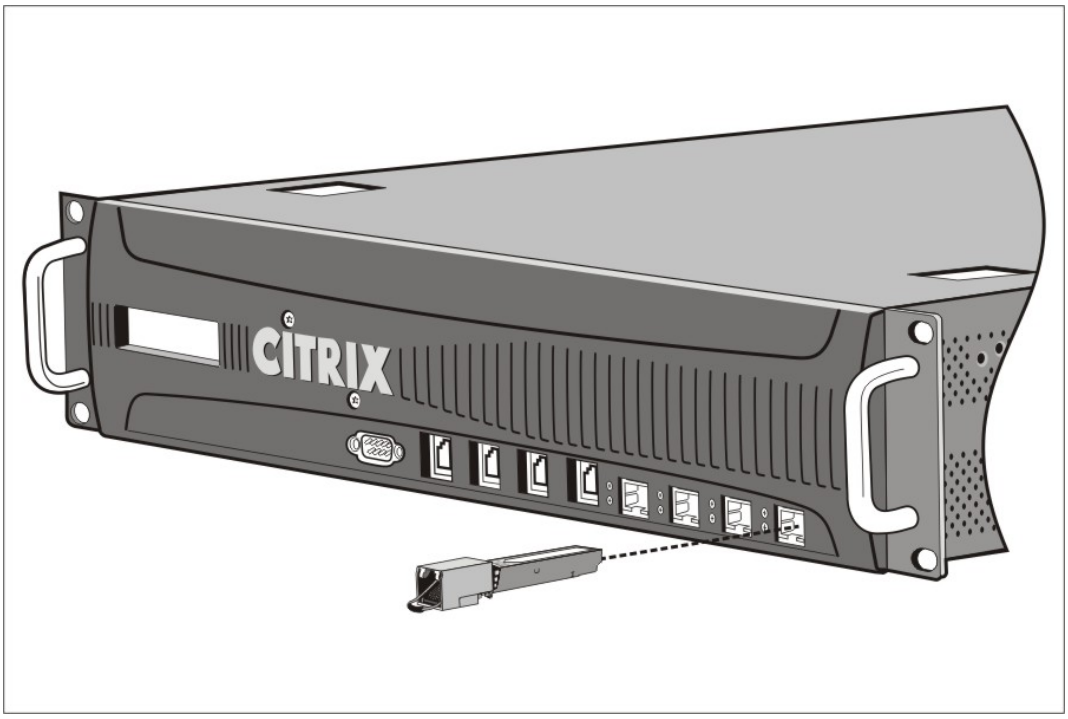


Figure 1. Installing an SFP transceiver

3. Hold the SFP transceiver between your thumb and index finger and insert it into the SFP transceiver port, pressing it in until you hear the transceiver snap into place.

4. Lock the transceiver.

5. Verify that the LED is green and blinks twice, which indicates that the transceiver is functioning correctly.

6. If you are using a fiber SFP transceiver, remove the dust caps attached to the transceiver and the cable only when you are ready to insert the cable.

To remove an SFP transceiver

1. Disconnect the cable from the SFP transceiver. If you are using a fiberoptic cable, replace the dust cap on the cable before putting it away. DANGERDo not look directly into fiberoptic transceivers or cables. They emit laser beams that can damage your eyes.

2. Unlock the SFP transceiver.

3. Hold the SFP transceiver between your thumb and index finger and slowly pull it out of the port.

4. If you are removing a fiber SFP transceiver, replace the dust cap before putting it away.

5. Put the SFP transceiver into its original box or another appropriate container.

Installing and Removing XFP and SFP+ Transceivers

Note: This section applies to the 12000 10G, MPX 9700/10500/12500/15500, MPX 15000, MPX 17000, MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances.

A 10-Gigabit Small Form Factor Pluggable (XFP or SFP+) is a compact optical transceiver that can operate at speeds of up to 10 gigabits per second. The 12000 10G, MPX 15000, and MPX 17000 appliances use XFP transceivers and the MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances use SFP+ transceivers. Auto-negotiation is enabled by default on the XFP/SFP+ ports into which you insert your XFP/SFP+ transceiver. As soon as a link between the port and the network is established, the mode is matched on both ends of the cable and for SFP+ transceivers, the speed is also autonegotiated.

Note: XFP and SFP+ transceivers are **not hot-swappable** on the NetScaler appliances. You must restart a NetScaler appliance after you insert a 10 GE XFP or SFP+ transceiver.

Caution: Only XFP/SFP+ transceivers provided by Citrix Systems are supported on NetScaler appliances. Attempting to install third-party XFP/SFP+ transceivers on your NetScaler appliance voids the warranty.

Insert the XFP/SFP+ transceivers into the XFP/SFP+ ports on the front panel of the appliance. Frequent installation and removal of transceivers shortens their life span. Follow the removal procedure carefully to avoid damaging the transceiver or the appliance.

Caution: Do not install the transceivers with the cables attached. Doing so can damage the cable, the connector, or the optical interface of the transceiver.

To install an XFP/SFP+ transceiver

1. Remove the XFP/SFP+ transceiver carefully from its box. DANGERDo not look directly into fiberoptic transceivers and cables. They emit laser beams that can damage your eyes.

2. Align the XFP/SFP+ transceiver to the front of the XFP/SFP+ transceiver port on the front panel of the appliance.

3. Hold the XFP/SFP+ transceiver between your thumb and index finger and insert it into the XFP/SFP+ transceiver port, pressing it in until you hear the transceiver snap into place.

4. Move the locking hinge to the DOWN position as shown in the following figure.

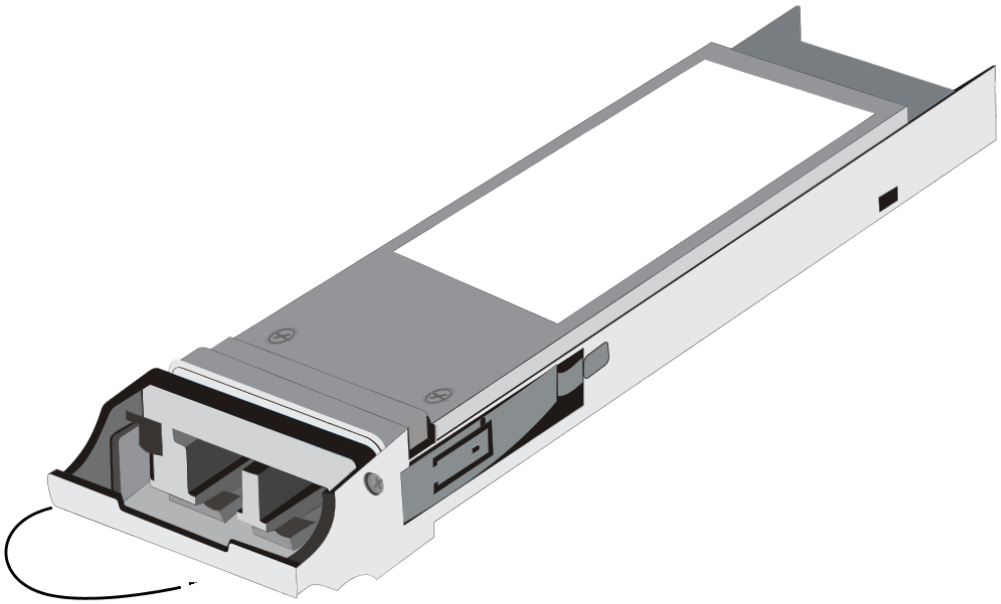


Figure 1. Locking an XFP transceiver

5. Verify that the LED is green and blinks twice, which indicates that the transceiver is functioning correctly.

6. Remove the dust caps attached to the transceiver and cable only when you are ready to insert the cable.

To remove an XFP/SFP+ transceiver

1. Disconnect the cable from the XFP/SFP+ transceiver. Replace the dust cap on the cable before putting it away. DANGERDo not look directly into fiberoptic transceivers or cables. They emit laser beams that can damage your eyes.

2. Unlock the XFP/SFP+ transceiver by moving the locking hinge to the UP position.

3. Hold the XFP/SFP+ transceiver between your thumb and index finger and slowly pull it out of the port.

4. Replace the dust cap on the transceiver before putting it away.

5. Put the XFP/SFP+ transceiver into its original box or another appropriate container.

Connecting the Cables

When the appliance is securely mounted on the rack, you are ready to connect the cables. Ethernet cables and the optional console cable are connected first. Connect the power cable last.

DANGERRemove all jewelry and other metal objects that might come in contact with power sources or wires before installing or repairing the appliance. When you touch both a live power source or wire and ground, any metal objects can heat up rapidly, and may cause burns, set clothing on fire, or fuse the metal object to an exposed terminal.

Connecting the Ethernet Cables

Ethernet cables connect your appliance to the network. The type of cable you need depends on the type of port used to connect to the network. Use a category 5e or category 6 Ethernet cable with a standard RJ-45 connector on a 10/100/1000BASE-T port or 1-gigabit SFP copper transceiver. Use a fiber optic cable with an LC duplex connector with an SFP fiber transceiver, SFP+, or XFP transceiver. The type of connector at the other end of the fiber optic cable depends on the port of the device that you are connecting to.

To connect an Ethernet cable to a 10/100/1000BASE-T port or 1-gigabit SFP copper transceiver

1. Insert the RJ-45 connector on one end of your Ethernet cable into an appropriate port on the front panel of the appliance, as shown in the following figure.

Figure 1. Inserting an Ethernet cable

2. Insert the RJ-45 connector on the other end into the target device, such as a router or switch.

3. Verify that the LED glows amber when the connection is established.

To connect the Ethernet cable to an SFP fiber, SFP+, or XFP transceiver

1. Remove the dust caps from the transceiver and cable.
2. Insert the LC connector on one end of the fiber optic cable into the appropriate port on the front panel of the appliance.
3. Insert the connector on the other end into the target device, such as a router or switch.
4. Verify that the LED glows amber when the connection is established.

Connecting the Console Cable

You can use the console cable to connect your appliance to a computer or terminal, from which you can configure the appliance. Alternatively, you can use a computer connected to the network. Before connecting the console cable, configure the computer or terminal to support VT100 terminal emulation, 9600 baud, 8 data bits, 1 stop bit, parity, and flow control set to NONE. Then connect one end of the console cable to the RS232 serial port on the appliance and the other end to the computer or terminal.

To connect the console cable to a computer or terminal

1. Insert the DB-9 connector at the end of the cable into the console port that is located on the front panel of the appliance as shown in the following figure.

   Figure 2. Inserting a console cable

   Note: To use a cable with an RJ-45 converter, insert the optional converter provided into the console port and attach the cable to it.

2. Insert the RJ-45 connector at the other end of the cable into the serial port of the computer or terminal.

Connecting the Power Cable

An MPX 5500, MPX 7500/9500, and 7000 appliance has one power cable. All the other appliances come with two power cables, but they can also operate if only one power cable is connected. A separate ground cable is not required, because the three-prong plug provides grounding.

To connect the appliance to the power source

1. Connect one end of the power cable to the power outlet on the back panel of the appliance, next to the power supply, as shown in the following figure.

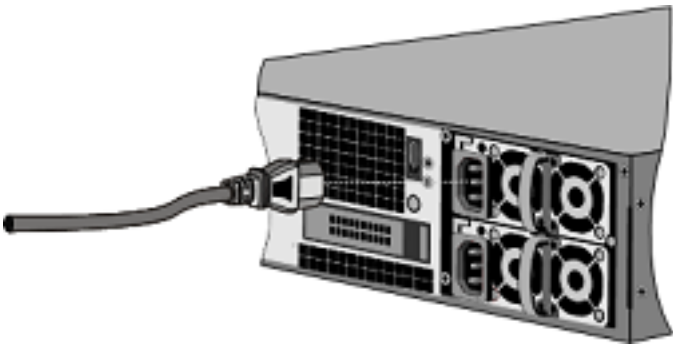


Figure 3. Inserting a power cable

2. Connect the other end of the power cable to a standard 110V/220V power outlet.

3. If a second power supply is provided, repeat steps 1 and 2 to connect the second power supply.

Note: The 9010, 10010, 12000, MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances emit a high-pitched alert if one power supply fails or if you connect only one power cable to the appliance. To silence the alarm, you can press the small red button located on the back panel of the appliance.

Turning on the Appliance

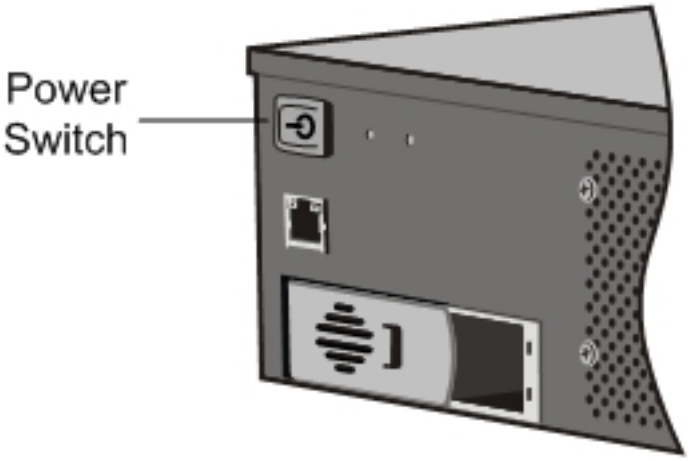
After you have installed the appliance in a rack and connected the cables, verify that the power cable is properly connected. When two power supplies are present, make sure the second cable is connected to an outlet for a different circuit than the first. After verifying the connections, you are ready to turn on the appliance.

To turn on the appliance

1. Verify that the appliance is connected through a console or Ethernet port. This will ensure that you can configure the appliance after it is turned on.

2. Press the ON/OFF toggle power switch on the back panel of the appliance.

Figure 1. Power switch on back panel



3. Verify that the LCD on the front panel is backlit and the start message appears, as shown in the following figure.

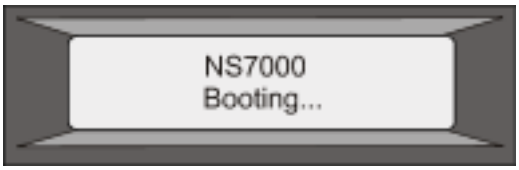


Figure 2. LCD startup screen

Caution: Be aware of the location of the emergency power off (EPO) switch. If an electrical accident occurs, you can quickly remove power from the appliance.

Initial Configuration

After you have installed your appliance in a rack, you are ready to perform the initial configuration. Once initial configuration is complete, you need to refer to the specific configuration guides for the features you will be using.

Initial configuration is the same for the multifunction Citrix NetScaler, the dedicated Citrix Access Gateway Enterprise Edition, and the dedicated Citrix Application Firewall appliances. To perform the initial configuration on the MPX 5500, MPX 7500/9500, MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances, you can also use the LCD keypad on the front panel of the appliance.To perform the initial configuration, you can use the serial console or the setup wizard. You can access the setup wizard from any computer that is on the same network as the new NetScaler. However, because this method uses the NetScaler default IP address, you must install and configure one NetScaler appliance at a time. If you want to configure a new NetScaler from a remote network, or if you want to install multiple NetScalers and then configure them without using the console port, you can use Dynamic Host Configuration Protocol (DHCP) to assign each new NetScaler an IP address at which you can access the appliance for remote configuration.

After you complete the initial configuration of the appliance, you can configure secure access to your appliance. As a result, you are no longer prompted for a password when logging on. This is especially helpful in environments for which you would otherwise have to keep track of a large number of passwords.

Using the LCD Keypad

Note: The following section is applicable to MPX 5500, MPX 7500/9500, MPX 9700/10500/12500/15500, MPX 11500/13500/14500/16500/18500/20500, MPX 17500/19500/21500, and MPX 17550/19550/20550/21550 appliances only.

When you first install the appliance, you can configure the initial settings by using the LCD keypad on the front panel of the appliance. The keypad interacts with the LCD display module, which is also on the front panel of these appliances.

Note: You can use the LCD keypad for initial configuration on a new appliance with the default configuration. The configuration file (ns.conf) should contain the following command and default values.

set ns config -**IPAddress** 192.168.100.1 **-netmask** 255.255.0.0

The functions of the different keys are explained in the following table.

Table 1. LCD Key Functions

Key	Function
<	Moves the cursor one digit to the left.
>	Moves the cursor one digit to the right.
^	Increments the digit under the cursor.
v	Decrements the digit under the cursor.
.	Processes the information, or terminates the configuration, if none of the values are changed. This key is also known as the ENTER key.

You are prompted to enter the subnet mask, NetScaler IP address (NSIP), and gateway in that order respectively. The subnet mask is associated with both the NSIP and default gateway IP address. The NSIP is the IPv4 address of the NetScaler appliance. The default gateway is the IPv4 address for the router, which will handle external IP traffic that the NetScaler cannot otherwise route. The NSIP and the default gateway should be on the same subnet.

If you enter a valid value for the subnet mask, such as 255.255.255.224, you are prompted to enter the IP address. Similarly, if you enter a valid value for the IP address, you are prompted to enter the gateway address. If the value you entered is invalid, the following error message appears for three seconds, where xxx.xxx.xxx.xxx is the IP address you entered, followed by a request to re-enter the value.

Invalid addr!
xxx.xxx.xxx.xxx

If you press the ENTER (.) key without changing any of the digits, the software interprets this as a user exit request.The following message will be displayed for three seconds.

Exiting menu...
xxx.xxx.xxx.xxx

If all the values entered are valid, when you press the ENTER key, the following message appears.

Values accepted,
Rebooting...

The subnet mask, NSIP, and gateway values are saved in the configuration file.

Using the NetScaler Serial Console

When you first install the appliance, you can configure the initial settings by using the serial console. With the serial console, you can change the system IP address, create a subnet or mapped IP address, configure advanced network settings, and change the time zone.

Note: To locate the serial console port on your appliance, see "RS232 Serial Console Port" in Ports.

To configure initial settings by using a serial console

1. Connect the console cable into your appliance. For more information, see "Connecting the Console Cable" in Connecting the Cables.

2. Run the vt100 terminal emulation program of your choice on your computer to connect to the appliance.

   - For Microsoft Windows, you can use HyperTerminal, which is installed with all current versions of Windows.

   - For Apple Macintosh OSX, you can use the GUI-based Terminal program or the shell-based telnet client.

     Note: OSX is based on the FreeBSD UNIX platform. Most standard UNIX shell programs are available from the OSX command line.

   - For UNIX-based workstations, you can use the shell-based telnet client or any supported terminal emulation program.

3. Press ENTER. The terminal screen displays the Logon prompt.

   Note: You might have to press ENTER two or three times, depending on which terminal program you are using.

4. Log on to the appliance with the administrator credentials. Your sales representative or Citrix Customer Service can provide you with the administrator credentials.

5. At the prompt, type config nsto run the NetScaler configuration script.

6. To complete the initial configuration of your appliance, follow the prompts.

   Note: To prevent an attacker from breaching your ability to send packets to the appliance, choose a non-routable IP address on your organization's LAN as your appliance IP address.

   You can replace steps 5 and 6 with the following NetScaler commands. At the NetScaler command prompt, type:

   set ns config **-ipaddress**<IPAddress> **-netmask**<subnetMask>

   add ns ip<IPAddress> <subnetMask> **-type**<type>

   add route<network> <subnetMask> <gateway>

   set system user<userName> <password>

   save ns config

   reboot

   Example

set ns config -ipaddress 10.102.29.60 -netmask 255.255.255.0 add ns ip 10.102.29.61 255.255.255.0 -typ

You have now completed initial configuration of your appliance. To continue configuring the appliance, choose one of the following options:

Citrix NetScaler.

If you are configuring your appliance as a standard NetScaler with other licensed features, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Citrix Application Firewall.

If you are configuring your appliance as a standalone application firewall, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Citrix Access Gateway.

If you are configuring your appliance as an Access Gateway, see Access Gateway 9.3, Enterprise Edition.

Note: For information about deploying a high availability (HA) pair, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671. For information about configuring a Federal Information Processing Standards (FIPS) appliance, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Using the Setup Wizard

To configure the appliance by using the Setup Wizard in the configuration utility, you need an administrative computer configured on the same network as the appliance. To run the configuration utility, the Java RunTime Environment (JRE) version 1.4.2_04 or later must be installed on the workstation or laptop. You can use the Setup Wizard to configure the following initial settings on the appliance:

- System IP address and subnet mask
- Subnet or Mapped IP address and subnet mask
- Host name
- Default gateway
- Time zone
- Licenses
- Administrator password

Important: Before running the Setup Wizard, you should download your licenses from the Citrix Web site and put them in a location on your computer or another device where you can access them from your Web browser during configuration.

Note: If the appliance is configured with the default IP address, licenses are not installed on the appliance, or the mapped or subnet IP address is not configured, the configuration utility automatically opens the Setup Wizard when you log on to the appliance.

To configure initial settings by using the Setup Wizard

1. In a Web browser, type: `http://192.168.100.1`

   Note: The operating system is preconfigured with a default IP address and associated netmask. The default IP address is 192.168.100.1 and the default netmask is 255.255.0.0.

2. In **User Name** and **Password**, type the administrator credentials. You can obtain the initial user name and password from your sales representative or from Citrix Customer Service.

3. In **Start in**, select **Configuration**, and then click **Login**.

4. In the **Setup Wizard**, click **Next**, and then follow the instructions in the wizard.

   Note: To prevent an attacker from breaching your ability to send packets to the appliance, choose a non-routable IP address on your organization's LAN as your appliance IP address.

   You have now completed initial configuration of your appliance. To continue configuring the appliance, choose one of the following options:

   Citrix NetScaler.

   If you are configuring your appliance as a standard NetScaler with other licensed features, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

   Citrix Application Firewall.

   If you are configuring your appliance as a standalone application firewall, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

   Citrix Access Gateway.

   If you are configuring your appliance as an Access Gateway, see Access Gateway 9.3, Enterprise Edition.

   Note: For information about deploying a high availability (HA) pair, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671. For information about configuring a Federal Information Processing Standards (FIPS) appliance, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Using DHCP for Initial Access

For initial configuration of a Citrix NetScaler appliance, Dynamic Host Configuration Protocol (DHCP) can eliminate dependency on the console by providing an IP address at which you can access the appliance to configure it remotely. You can also use DHCP after initial configuration if, for example, you want to move a NetScaler to a different subnet.

To use DHCP, you must first specify the NetScaler vendor class identifier on a DHCP server. Optionally, you can also specify the pool of IP addresses from which your NetScaler appliance can acquire an IP address. If a pool is not specified, the address is acquired from the general pool.

A new NetScaler does not have a configuration file. When you connect a NetScaler without a configuration file to the network, its DHCP client automatically polls the DHCP server for an IP address. If you have specified the NetScaler vendor class identifier on the DHCP server, the server returns an address. You can also enable the DHCP client on a previously configured NetScaler.

Prerequisites

To use DHCP, you must:

1. Note the system ID (sysid) on the serial number sticker on the back panel of the appliance. On an older NetScaler, the system ID may not be available. In this case, use the MAC address instead of the system ID.

2. Set up a DHCP server and configure it with the NetScaler vendor class identifier.

To configure a Linux/UNIX DHCP server for the NetScaler

1. Specify "citrix-NS" as the vendor class identifier for the NetScaler by adding the following configuration to the server's dhcpd.conf file:

   ```
   subclass "citrix-1" "citrix-NS"{
   vendor-option-space auto;
   option auto.key "citrix-NS";
   ```

   Note: The location of the dhcpd.conf file can be different in different versions and flavors of the Linux/UNIX-based operating system (for example, in FreeBSD 6.3 the file is present in the /etc/ folder). For the location, see the `dhcpd` man page of the DHCP server.

2. If you do not want NetScaler appliances to use IP addresses from the general pool, specify a pool of addresses for the NetScaler. For example, adding the following configuration to the dhcpd.conf file specifies a pool of IP addresses ranging from 10.102.33.246 to 10.102.33.249.

   ```
   pool {
   allow members of "citrix-1";
   range 10.102.33.246 10.102.33.249;
   option subnet-mask 255.255.255.0;
   }
   ```

3. Terminate the DHCP process and restart it to reflect the change to the configuration file. At the shell prompt, type:

   ```
   killall dhcpd

   dhcpd&
   ```

Implementing an Initial NetScaler Configuration from a Remote Computer

When a new NetScaler (or any NetScaler that does not have a configuration file) starts, it automatically polls the DHCP server for an IP address and provides the DHCP server with its sysid. The DHCP server includes this sysid with the IP address that it assigns to the NetScaler in the server's dhcpd.leases file. To find the IP address currently assigned to your NetScaler, look in the dhcpd.leases file for the last entry with the sysid of your NetScaler in the uid or client-hostname field. Verify that the binding state in this entry is active. If the binding state is not active but free, the IP address is not yet associated with the NetScaler.

You can use this address to connect to the NetScaler and remotely configure the initial settings. For example, you can change the IP address, subnet mask, and gateway settings that were fetched from the DHCP server. After completing the initial configuration, you can manually return the DHCP IP address to the server pool. Alternatively, restarting the NetScaler automatically releases the DHCP IP address back to the server pool. A restart also saves the NetScaler configuration file.

Example

The following code is an example of an entry in a DHCP server's dhcpd.leases file. This entry verifies the binding state of the NetScaler whose sysid is 45eae1a8157e89b9314f.

```
lease 10.102.33.248 {
  starts 3 2009/08/19 00:40:37;
  ends 3 2009/08/19 06:40:37;
  cltt 3 2009/08/19 00:40:37;
  binding state active;
  next binding state free;
  hardware ethernet 00:d0:68:11:f4:d6;
  uid "45eae1a8157e89b9314f";
  client-hostname "45eae1a8157e89b9314f";
```

In the above example, the binding state is ACTIVE and the IP address assigned to the NetScaler is 10.102.33.248.

The following table describes DHCP-related CLI commands that you might want to use when configuring a new NetScaler.

Table 1. NetScaler CLI commands for using DHCP with a new NetScaler

Task	At the NetScaler command prompt, type:
To verify the DHCP fetched details, such as IP address, subnet mask, and gateway on the NetScaler	`> sh dhcpParams`
To release the DHCP IP address and return it to the IP address pool on the DHCP server when the NetScaler configuration is complete	`> release dhcpIP`

Using DHCP When a Configuration File is Present

If you need to move a NetScaler to a different subnet, such as from a testing environment to a production environment, you can use DHCP to access a NetScaler that already has a configuration file. Before moving the NetScaler, enable its DHCP client and save the configuration. As a result, when the NetScaler restarts, it automatically polls the DHCP server for an IP address. If you did not enable the DHCP client and save the configuration before shutting down the NetScaler, you will need to connect to the NetScaler through the console and dynamically run the DHCP client on the NetScaler. The DHCP server will then provide an IP address, a gateway, and a subnet mask. You can use the IP address to access the NetScaler and configure the other settings remotely.

If the DHCP client is enabled in the configuration file, you should disable it and then save the configuration file. If the DHCP client is enabled, the NetScaler will poll the DHCP server again for an IP address when it restarts.

The following table lists the NetScaler CLI commands associated with each task.

Table 2. NetScaler CLI commands for using DHCP with a previously configured NetScaler

Task	At the NetScaler command prompt, type:
To dynamically run the DHCP client to fetch an IP address from the DHCP server	`> set dhcpParams dhcpClient on`

To configure the DHCP client to run when the NetScaler restarts	`> set dhcpParams dhcpClient on`
	
	`> save config`

To prevent the DHCP client from running when the NetScaler restarts	`> set dhcpParams dhcpClient off`
	
	`> save config`
	
	Note: This is required only if the ON setting was saved.

To save the DHCP acquired route so that it is available when the NetScaler restarts	`> set dhcpParams -dhcpclient on -saveroute on`
	
	`> save config`

To prevent saving the DHCP acquired route (default behavior)	`> set dhcpParams -dhcpclient on -saveroute off`
	
	`> save config`
	
	Note: This is required only if the ON setting was saved.

Accessing a NetScaler by Using SSH Keys and No Password

In a setup with a large number of NetScaler appliances, you will have to store and look up passwords for each appliance before you can log on to the appliance. To avoid this, you can set up secure shell access with public key encryption on the appliance so that you are not prompted for the password. To do this, you will need to first generate the public/private key on the client and then copy the public key to the NetScaler.

To generate the public/private key on a Linux client

1. Change directory to /root/.ssh
2. Generate the public and private key pair. At the prompt, type:

   [root@localhost .ssh]# ssh-keygen **-t** rsa
3. Press ENTER when prompted for a file name to save the key.
4. Press ENTER when prompted for a passphrase.

To copy the public key to the remote NetScaler

1. Log on to the remote NetScaler from the Linux client.
2. Change directory to /nsconfig/ssh. At the prompt, type:

   cd /nsconfig/ssh
3. Change to binary mode and copy the public key to this directory. At the prompt, type:

   bin

   put id_rsa.pub

To set up secure shell access with public key encryption on the NetScaler

1. Open a connection to the NetScaler using an SSH client, such as PuTTY.
2. Log on to the NetScaler with the administrator credentials.
3. At the shell prompt, change the directory to /nsconfig/ssh.
4. Append the public key to the authorized_keys file and change permissions. At the prompt, type:

   ```
   cat id_rsa.pub >> authorized_keys

   chmod 755 authorized_keys
   ```

5. Remove the public key (optional). At the prompt, type:

   ```
   rm id_rsa.pub
   ```

6. At the prompt type the following command to complete the configuration:

   cp authorized_keys /root/.ssh/authorized_keys2

7. Change the directory to /nsconfig. At the prompt type:

   ```
   cd /nsconfig
   ```

8. To prevent your changes from being lost if the NetScaler is restarted add the following line to the rc.netscaler file:

   ```
   cp /nsconfig/ssh/authorized_keys /root/.ssh/authorized_keys2
   ```

   Important: If the /nsconfig directory contains no rc.netscaler file, you must create one.

To verify secure shell access with public key encryption on the NetScaler

On the client, verify that you can connect to the remote NetScaler by using SSH, without entering the password. At the prompt, type:

ssh nsroot@<NSIPaddress>

You should not be prompted for a password.

Example

```
ssh nsroot@10.102.96.50
```

Lights Out Management on the NetScaler Appliance

The MPX 11500/13500/14500/16500/18500/20500 and MPX 17550/19550/20550/21550 appliances have an Intelligent Platform Management Interface (IPMI) also known as the Lights out Management (LOM) port on the front panel of the appliance. Using LOM, you can remotely monitor and manage the appliance independently of the operating system. You can remotely change the IP address, power cycle the appliance, and perform a code dump by connecting to the appliance through the LOM port.

By connecting the LOM port over a dedicated channel that is separate from the data channel, you can make sure that connectivity to the appliance is maintained even if the data network is down.

Configuring the LOM Port

For initial configuration of the LOM port, connect to the port's default IP address and change it to the address that you want to use for remote monitoring and management. Also specify the administrator credentials and the network settings.

Note: The LEDs on the LOM port are unoperational by design.

To configure the LOM port

1. In a Web browser, type the IP address of the LOM port. For initial configuration, type the port's default address: `http://192.168.1.3`
2. In the **User Name** and **Password** boxes, type the administrator credentials. You can obtain the initial user name and password from your sales representative or from Citrix Customer Service.
3. In the Menu bar, click **Configuration.**
4. Under **Options**, click **Network** and type values for the following parameters:
   - IP Address—The IP address of the LOM port.
   - Subnet Mask—The mask used to define the subnet of the LOM port.
   - Default Gateway—The IP address of the router that connects the appliance to the network.
5. Click **Save.**

Power Cycling the Appliance

You can remotely turn off the appliance and turn it back on. The result is similar to pressing the power button on the back panel of the appliance for less than four seconds. The operating system performs a graceful shutdown. All operations on the appliance are stopped, no new connections to the client or server are accepted, and all the existing connections are closed.

To power cycle the appliance

1. In a Web browser, type the IP address of the LOM port.
2. In the **User Name** and **Password** boxes, type the administrator credentials.
3. In the **Menu** bar, click **Remote Control**.
4. Under **Options**, click **Power Control**, and then click **Power Cycle Server**.
5. Click **Perform Action**.

Performing a Core Dump

If the appliance fails or becomes unresponsive, you can remotely perform a core dump. This procedure has the same effect as pressing the NMI button on the back panel of the appliance.

To perform a core dump

1. In a Web browser, type the IP address of the LOM port.
2. In the **User Name** and **Password** boxes, type the administrator credentials.
3. In the **Menu** bar, click **Remote Control**.
4. Under **Options**, click **Power Control**, and then click **NMI Dump**.
5. Click **Perform Action**.

Administration

The following topics provide a conceptual reference and instructions for managing and monitoring the Citrix® NetScaler® appliance by using built-in features, such as command policies, Simple Network Management (SNMP), audit server logging, web server logging, Network Time Protocol (NTP), and the Reporting tool.

Citrix NetScaler Authentication and Authorization	Configure authentication and authorization to manage access to the NetScaler and different parts of the NetScaler configuration.
SNMP	Learn how SNMP works with NetScaler and how to configure SNMP V1, V2, and V3 on NetScaler.
Audit Server Logging	Configure the NetScaler audit server log to log and monitor the NetScaler states and status information. Also, learn how to configure audit server logging on a server system and for a deployment scenario.
Web Server Logging	Configure web server log to maintain a history of the page requests that originate from the NetScaler.
Advanced Configurations	Learn how to set advanced configurations, such as NTP, PMTU, and autodetected services, on the NetScaler.
Web Interface	Learn how to configure Web Interface on the NetScaler appliance for providing access to Citrix® XenApp™ and Citrix® XenDesktop® applications.
Enhanced Application Visibility Using AppFlow	Learn how to configure AppFlow for collecting network flow information.
Reporting Tool	Learn how to use the Reporting tool to view performance statistics as reports with graphs that are based on statistics collected by the nscollect utility.

Authentication and Authorization

To configure Citrix® NetScaler® authentication and authorization, you must first define the users who have access to the NetScaler appliance, and then you can organize these users into groups. After configuring users and groups, you need to configure command policies to define types of access, and assign the policies to users and/or groups.

You must log on as an administrator to configure users, groups, and command policies. The default NetScaler administrator user name is *nsroot*. After logging on as the default administrator, you should change the password for the nsroot account. Once you have changed the password, no user can access the NetScaler appliance until you create an account for that user. If you forget the administrator password after changing it from the default, you can reset it to nsroot.

Authentication and Authorization

To configure Citrix® NetScaler® authentication and authorization, you must first define the users who have access to the NetScaler appliance, and then you can organize these users into groups. After configuring users and groups, you need to configure command policies to define types of access, and assign the policies to users and/or groups.

You must log on as an administrator to configure users, groups, and command policies. The default NetScaler administrator user name is *nsroot*. After logging on as the default administrator, you should change the password for the nsroot account. Once you have changed the password, no user can access the NetScaler appliance until you create an account for that user. If you forget the administrator password after changing it from the default, you can reset it to nsroot.

Configuring Users and Groups

You must define your users by configuring accounts for them. To simplify the management of user accounts, you can organize them into groups.

You can also customize the NetScaler command-line prompt for a user. Prompts can be defined in a user's configuration, in a user-group configuration, and in the global configuration. The prompt displayed for a given user is determined by the following order of precedence:

1. Display the prompt as defined in the user's configuration.
2. Display the prompt as defined in the group configuration for the user's group.
3. Display the prompt as defined in the system global configuration.

Configuring User Accounts

To configure user accounts, you simply specify user names and passwords. You can change passwords and remove user accounts at any time.

To create a user account by using the NetScaler command line

At the NetScaler command prompt, type the following command to create a user account and verify the configuration:

- add system user <userName> [-**promptString** <string>]
- show system user

Example

```
> add system user user1
Enter password:
Confirm password:
 Done

> add system user johnd -promptString user-%u-at-%T
Enter password:
Confirm password:
 Done
```

```
> show system user
1)      User name: nsroot
2)      User name: user1
3)      User name: johnd  Prompt String: user-%u-at-%T  Prompt Inherited From: User
 Done
```

To modify or remove a user account by using the NetScaler command line

- To modify a user's password, type the set system user <userName> command and the parameters to be changed, with their new values.
- To remove a user account, type the rm system user <userName> command.

Parameters for configuring a user account

userName

A name for the user. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

password

A password that the user uses to log on.

promptString

A name for the user’s NetScaler command-line prompt. The name can consist of letters, numbers, the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), underscore (_) symbols, and the following variables:

- **%u**—Will be replaced by the user name.
- **%h**—Will be replaced by the host name of the NetScaler appliance.
- **%t**—Will be replaced by the current time in 12-hour format.
- **%T**—Will be replaced by the current time in 24-hour format.
- **%d**—Will be replaced by the current date.
- **%s**—Will be replaced by the state of the NetScaler appliance.

A maximum of 63 characters are allowed for this parameter. A variable (for example, %u) is counted as two characters. The resulting prompt can be longer than 63 characters.

To configure a user account by using the configuration utility

1. In the navigation pane, expand **System** and click **Users**.
2. In the details pane, do one of the following:
   - To create a user account, click **Add**.
   - To modify an existing user account, select the user, and then click **Open**.
3. In the **Create System User** or **Configure System User** dialog box, specify values for the parameters, which correspond to parameters described in "Parameters for configuring a user account" as shown:
   - **User Name*** —userName (Cannot be changed for an existing user.)
   - **Password*** —password
   - **Confirm Password*** —password
   - **CLI Prompt**—promptString (Optional)

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the user has been configured successfully.

Configuring User Groups

After configuring a user group, you can easily grant the same access rights to everyone in the group. To configure a group, you create the group and bind users to the group. You can bind each user account to more than one group. Binding user accounts to multiple groups may allow more flexibility when applying command policies.

To create a user group by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a user group and verify the configuration:

- add system group <groupName> [-**promptString** <string>]
- show system group

Example

```
> add system group Managers -promptString Group-Managers-at-%h
 Done
> show system group
1)      Group name: group1
2)      Group name: Managers  Prompt String: Group-Managers-at-%h
 Done
```

To modify or remove a user group by using the NetScaler command line

- To modify a user group, type the set system group <groupName> command and the parameters to be changed, with their new values.
- To remove a user group, type rm system group <groupName>.

To bind a user to a group by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a user account to a group and verify the configuration:

- bind system group <groupName> **-userName** <userName>
- show system group <groupName>

Example

```
> bind system group Managers -userName user1
 Done

> bind system group Managers -userName johnd
 Done

> show system group Managers
      Group name: Managers  Prompt String: Group-Managers-at-%h
      User name: user1
      User name: johnd
Done

> show system user user1
User name: user1  Prompt String: Group-Managers-at-%h  Prompt Inherited From: Group

      Group name: Managers
 Done

> show system user johnd
User name: johnd  Prompt String: user-%u-at-%T  Prompt Inherited From: User

      Group name: Managers
 Done
```

To unbind a user from a group by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a user account and verify the configuration:

- unbind system group <groupName> **-userName** <userName>
- show system group <groupName>

Parameters for configuring a user group

groupName

A name for the group you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing groups.)

userName

The name that was assigned to a previously configured user.

promptString

A name for the NetScaler command-line prompt for all the users that are part of this group. The name can consist of letters, numbers, the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), underscore (_) symbols, and the following variables:

- **%u**—Will be replaced by the user name.
- **%h**—Will be replaced by the host name of the NetScaler appliance.
- **%t**—Will be replaced by the current time in 12-hour format.
- **%T**—Will be replaced by the current time in 24-hour format.
- **%d**—Will be replaced by the current date.
- **%s**—Will be replaced by the state of the NetScaler appliance.

A maximum of 63 characters are allowed for this parameter. A variable (for example, %u) is counted as two characters. The resulting prompt can be longer than 63 characters.

To configure a user group by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Groups**.

2. In the details pane, do one of the following:

   - To create a new user group, click **Add**.

   - To modify an existing user group, select the group, and then click **Open**.

3. In the **Create System Group** or **Configure System Group** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a user group" as shown:

   - **Group Name*****—groupName (Required for a new group. Cannot be changed for an existing group.)

   - **CLI Prompt**—promptString (Optional)

   * A required parameter

4. Under **Members**, select users from the **Available Users** list and click **Add** to move them to the **Configured Users** list.

5. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the group has been configured successfully.

Configuring Command Policies

Command policies regulate which commands, command groups, vservers, and other entities that users and user groups are permitted to use.

The Citrix® NetScaler® appliance provides a set of built-in command policies, and you can configure custom policies. To apply the policies, you bind them to users and/or groups.

Here are the key points to keep in mind when defining and applying command policies.

- You cannot create global command policies. Command policies must be bound directly to NetScaler users and groups.
- Users or groups with no associated command policies are subject to the default (DENY-ALL) command policy, and are therefore unable to execute any configuration commands until the proper command policies are bound to their accounts.
- All users inherit the policies of the groups to which they belong.
- You must assign a priority to a command policy when you bind it to a user account or group account. This enables the NetScaler to determine which policy has priority when two or more conflicting policies apply to the same user or group.
- The following commands are available by default to any user and are unaffected by any command you specify:

  help cli, show cli attribute, clear cli prompt, alias, unalias, help, history, quit, exit, whoami, config, set cli mode, unset cli mode, show cli mode, set cli prompt, and show cli prompt.

Built-in Command Policies

The following table describes the built-in policies.

Table 1. Built-in Command Policies

Policy name	Allows
read-only	Read-only access to all show commands except show runningconfig, show ns.conf, and the show commands for the NetScaler command group.
operator	Read-only access and access to commands to enable and disable services and servers or place them in ACCESSDOWN mode.

network	Full access, except to the set and unset SSL commands, sh ns.conf, sh runningconfig, and sh gslb runningconfig commands.
superuser	Full access. Same privileges as the nsroot user.

Creating Custom Command Policies

Regular expression support is offered for users with the resources to maintain more customized expressions, and for those deployments that require the flexibility that regular expressions offer. For most users, the built-in command policies are sufficient. Users who need additional levels of control but are unfamiliar with regular expressions may want to use only simple expressions, such as those in the examples provided in this section, to maintain policy readability.

When you use a regular expression to create a command policy, keep the following in mind.

- When you use regular expressions to define commands that will be affected by a command policy, you must enclose the commands in double quotation marks. For example, to create a command policy that includes all commands that begin with show, type the following:

  ```
  "^show .*$"
  ```

  To create a command policy that includes all commands that begin with rm, type the following:

  ```
  "^rm .*$"
  ```

- Regular expressions used in command policies are not case sensitive.

The following table lists examples of regular expressions:

Table 2. Examples of Regular Expressions for Command Policies

Command specification	Matches these commands
"^rm\s+.*$"	All remove actions, because all remove actions begin with the rm string, followed by a space and additional parameters and flags.
"^show\s+.*$"	All show commands, because all show actions begin with the show string, followed by a space and additional parameters and flags.
"^shell$"	The shell command alone, but not combined with any other parameters or flags.

"^add\s+vserver\s+.*$"	All create vserver actions, which consist of the add vserver command followed by a space and additional parameters and flags.
"^add\s+(lb\s+vserver)\s+.*"	All create lb vserver actions, which consist of the add lb vserver command followed by a space and additional parameters and flags.

The following table shows the command specifications for each of the built-in command policies.

Table 3. Expressions Used in the Built-in Command Policies

Policy name	Command specification regular expression
read-only	(^man.*)|(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)
operator	(^man.*)|(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)|(^set.*-accessdown.*)|(^(enable|disable) (server|service).*)
network	^(?!shell)\S+\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*
superuser	.*

To create a command policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a command policy and verify the configuration:

- add system cmdPolicy <policyname> <action> <cmdspec>
- sh system cmdPolicy

Example

```
> add system cmdPolicy read_all ALLOW (^show\s+(!system)(!ns ns.conf)(!ns runningConfig).*)|(^stat.*)
 Done
> sh system cmdPolicy
1)      Command policy: operator
2)      Command policy: read-only
3)      Command policy: network
4)      Command policy: superuser
5)      Command policy: allow_portaladmin
6)      Command policy: read_all
 Done
```

To modify or remove a command policy by using the NetScaler command line

- To modify a command policy, type the set system cmdPolicy <PolicyName> command and the parameters to be changed, with their new values.
- To remove a command policy, type rm system cmdPolicy <PolicyName>.

  Note: The built-in command policies cannot be removed.

Parameters for configuring a command policy

policyname

A name for the command policy you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing policies.)

action

The action the policy applies when the command specification pattern matches. Possible values: ALLOW, DENY

cmdspec

Rule (expression) that the policy uses for pattern matching.

To configure a command policy by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Command Policies**.
2. In the details pane, do one of the following:
   - To create a command policy, click **Add**.
   - To modify an existing command policy, select the command policy, and then click **Open**.
3. In the **Create Command Policy** or **Configure Command Policy** dialog box, specify values for the parameters, which correspond to the parameters described in "Parameters for configuring a command policy" as shown:
   - Policy Name*—policyname (Cannot be changed for an existing policy.)
   - Action—action
   - Command Spec*—cmdspec (You can type a complete expression directly into the text area, or you can click **Add** or **Regex Tokens** for assistance.The **Add** icon opens the **Add Command** dialog box, in which you can select a NetScaler entity and then select an operation to perform on the entity. The **Regex Tokens** icon displays regular expression tokens, which you can add to your expression by selecting them.)

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the command policy has been configured successfully.

Binding Command Policies to Users and Groups

Once you have defined your command policies, you must bind them to the appropriate user accounts and groups.

When you bind a policy, you must assign it a priority so that the NetScaler appliance can determine which command policy to follow when two or more applicable command policies are in conflict.

Command policies are evaluated in the following order:

- Command policies bound directly to users and the corresponding groups are evaluated according to priority number. A command policy with a lower priority number is evaluated before one with a higher priority number. Therefore, any privileges the lower-numbered command policy explicitly grants or denies are not overridden by a higher-numbered command policy.
- When two command policies, one bound to a user account and other to a group, have the same priority number, the command policy bound directly to the user account is evaluated first.

To bind command policies to a user by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a command policy to a user and verify the configuration:

- bind system user <userName> -**policyName** <policyName> <priority>
- sh system user <userName>

Example

```
> bind system user user1 -policyName read_all 1
Done
> sh system user user1
User name: user1

      Command Policy: read_all       Priority:1
 Done
```

To unbind command policies from a user by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a command policy from a user and verify the configuration:

- unbind system user <userName> **-policyName** <policyName>
- sh system user <userName>

Parameters for binding a command policy to a user

userName

The name of an existing user account.

policyName

The name of an existing command policy.

priority

The priority assigned to this policy.

To bind command policies to a user by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Users**.
2. In the details pane, select the user to which you want to bind a command policy, and then click **Open**.
3. In the **Configure System User** dialog box, under **Command Policies**, all of the command policies configured on your NetScaler appear on the list. Select the check box next to the name of the policy you want to bind to this user.
4. In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
5. Click **OK**. A message appears in the status bar, stating that the user has been configured successfully.

To bind command policies to a group by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a command policy to a user group and verify the configuration:

- bind system group <groupName> -**policyName** <policyName> <priority>
- sh system group <groupName>

Example

```
> bind system group Managers -policyName read_all 1
Done
> sh system group Managers
      Group name: Managers

      User name: johnd

      Command policy: read_all       Priority:1
 Done
```

To unbind command policies from a group by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a command policy from a user group and verify the configuration:

- unbind system group <groupName> -**policyName** <policyName>
- sh system group <groupName>

Parameters for binding a command policy to a group

groupName

The name of an existing user group.

policyName

The name of an existing command policy.

priority

The priority assigned to this command policy.

To bind command policies to a group by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Groups**.
2. In the details pane, select the group to which you want to bind a command policy, and then click **Open**.
3. In the **Configure System Group** dialog box, under **Command Policies**, all the command policies configured on your NetScaler appear on the list. Select the check box next to the name of the policy you want to bind to this group.
4. In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
5. Click **OK**. A message appears in the status bar, stating that the group has been configured successfully.

Resetting the Default Administrator (nsroot) Password

The nsroot account provides complete access to all features of the Citrix® NetScaler® appliance. Therefore, to preserve security, the nsroot account should be used only when necessary, and only individuals whose duties require full access should know the password for the nsroot account. Frequently changing the nsroot password is advisable. If you lose the password, you can reset it to the default and then change it.

To reset the nsroot password, you must boot the NetScaler into single user mode, mount the file systems in read/write mode, and remove the set NetScaler user nsroot entry from the ns.conf file. You can then reboot, log on with the default password (nsroot), and choose a new password.

To reset the nsroot password

1. Connect a computer to the NetScaler serial port and log on.

   Note: You cannot log on by using ssh to perform this procedure; you must connect directly to the NetScaler.

   As the operating system starts, it displays the following message:

   ```
   Hit [Enter] to boot immediately, or any other key for command
   prompt.

   Booting [kernel] in # seconds.
   ```

2. Press CTRL+C.

   The following message appears:

   Type '?' for a list of commands, 'help' for more detailed help.

   ok

3. Type **boot -s** and press the ENTER key to start the NetScaler in single user mode.

   After the NetScaler boots, it displays the following message:

   Enter full path name of shell or RETURN for /bin/sh:

4. Press the ENTER key to display the # prompt, and type the following commands to mount the file systems:

   fsck /dev/ad0s1a

   mount /dev/ad0s1a /flash

5. Using a text editor of your choice, edit the /flash/nsconfig/ns.conf file and remove the set system user nsroot entry.

6. Save the file and exit the text editor.

7. Type **reboot** and press the ENTER key to reboot the NetScaler.

   When the NetScaler completes rebooting, it prompts for the user name and password.

8. Log on as nsroot, with the password nsroot.

   Once logged on to the NetScaler, you will be required to enter a new nsroot user password.

9. Follow the prompts to change the password.

10. Exit the **config ns** menu.

Example of a User Scenario

The following example shows how to create a complete set of user accounts, groups, and command policies and bind each policy to the appropriate groups and users. The company, Example Manufacturing, Inc., has three users who will access the Citrix® NetScaler® appliance:

John Doe. The IT manager. John needs to be able to see all parts of the NetScaler configuration but does not need to modify anything.

- **Maria Ramiez.** The lead IT administrator. Maria needs to be able to see and modify all parts of the NetScaler configuration except for NetScaler commands (which local policy dictates must be performed while logged on as nsroot).
- **Michael Baldrock.** The IT administrator in charge of load balancing. Michael needs to be able to see all parts of the NetScaler configuration, but needs to modify only the load balancing functions.

The following table shows the breakdown of network information, user account names, group names, and command policies for the sample company.

Table 1. Sample Values for Creating Entities

Field	Value	Note
NetScaler host name	ns01.example.net	N/A
User accounts	johnd, mariar, and michaelb	John Doe, IT manager, Maria Ramirez, IT administrator and Michael Baldrock, IT administrator.
Groups	Managers and SysOps	All managers and all IT administrators.
Command Policies	read_all, modify_lb, and modify_all	Allow complete read-only access, Allow modify access to load balancing, and Allow complete modify access.

The following description walks you through the process of creating a complete set of user accounts, groups, and command policies on the NetScaler appliance named ns01.example.net.

The description includes procedures for binding the appropriate user accounts and groups to one another, and binding appropriate command policies to the user accounts and groups.

This example illustrates how you can use prioritization to grant precise access and privileges to each user in the IT department.

The example assumes that initial installation and configuration have already been performed on the NetScaler.

Configuration steps

1. Use the procedure described in Configuring User Accounts to create user accounts **johnd**, **mariar**, and **michaelb**.
2. Use the procedure described in Configuring User Groups to create user groups **Managers** and **SysOps**, and then bind the users **mariar** and **michaelb** to the **SysOps** group and the user **johnd** to the **Managers** group.
3. Use the procedure described in Creating Custom Command Policies to create the following command policies:
   - **read_all** with action **Allow** and command spec "(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)"
   - **modify_lb** with action as **Allow** and the command spec **"^set\s+lb\s+.*$"**
   - **modify_all** with action as **Allow** and the command spec **"^\S+\s+(?!system).*"**
4. Use the procedure described in Binding Command Policies to Users and Groups to bind the **read_all** command policy to the **SysOps** group, with priority value **1**.
5. Use the procedure described in Binding Command Policies to Users and Groups to bind the **modify_lb** command policy to user **michaelb**, with priority value **5**.

The configuration you just created results in the following:

- John Doe, the IT manager, has read-only access to the entire NetScaler configuration, but he cannot make modifications.
- Maria Ramirez, the IT lead, has near-complete access to all areas of the NetScaler configuration, having to log on only to perform NetScaler-level commands.
- Michael Baldrock, the IT administrator responsible for load balancing, has read-only access to the NetScaler configuration, and can modify the configuration options for load balancing.

The set of command policies that applies to a specific user is a combination of command policies applied directly to the user's account and command policies applied to the group(s) of which the user is a member.

Each time a user enters a command, the operating system searches the command policies for that user until it finds a policy with an ALLOW or DENY action that matches the command. When it finds a match, the operating system stops its command policy search and allows or denies access to the command.

If the operating system finds no matching command policy, it denies the user access to the command, in accordance with the NetScaler appliance's default deny policy.

Note: When placing a user into multiple groups, take care not to cause unintended user command restrictions or privileges. To avoid these conflicts, when organizing your users in groups, bear in mind the NetScaler command policy search procedure and policy ordering rules.

Configuring External User Authentication

External user authentication is the process of authenticating the users of the Citrix® NetScaler® appliance by using an external authentication server. The NetScaler supports LDAP, RADIUS, TACACS+, and NT4 authentication servers. To configure external user authentication, you must create authentication policies. You can configure one or many authentication policies, depending on your authentication needs. An authentication policy consists of an expression and an action. Authentication policies use NetScaler classic expressions, which are described in detail in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/ctx128673.

After creating an authentication policy, you bind it to the system global entity and assign a priority to it. You can create simple server configurations by binding a single authentication policy to the system global entity. Or, you can configure a cascade of authentication servers by binding multiple policies to the system global entity. If no authentication policies are bound to the system, users are authenticated by the onboard system.

Note: User accounts must be configured on the NetScaler appliance before users can be externally authenticated. You must first create an onboard system user for all users who will access the appliance, so that you can bind command policies to the user accounts. Regardless of the authentication source, users cannot log on if they are not granted sufficient command authorization through command policies bound to their user accounts or to a group of which they are a member.

Configuring LDAP Authentication

You can configure the NetScaler to authenticate user access with one or more LDAP servers. LDAP authorization requires identical group names in Active Directory, on the LDAP server, and on the NetScaler. The characters and case must also be the same.

By default, LDAP authentication is secured by using SSL/TLS protocol. There are two types of secure LDAP connections. In the first type, the LDAP server accepts the SSL/TLS connection on a port separate from the port used to accept clear LDAP connections. After users establish the SSL/TLS connection, LDAP traffic can be sent over the connection. The second type allows both unsecure and secure LDAP connections and is handled by a single port on the server. In this scenario, to create a secure connection, the client first establishes a clear LDAP connection. Then the LDAP command StartTLS is sent to the server over the connection. If the LDAP server supports StartTLS, the connection is converted to a secure LDAP connection by using TLS.

The port numbers for LDAP connections are:

- 389 for unsecured LDAP connections
- 636 for secure LDAP connections
- 3268 for Microsoft unsecure LDAP connections
- 3269 for Microsoft secure LDAP connections

LDAP connections that use the StartTLS command use port number 389. If port numbers 389 or 3268 are configured on the NetScaler, it tries to use StartTLS to make the connection. If any other port number is used, connection attempts use SSL/TLS. If StartTLS or SSL/TLS cannot be used, the connection fails.

When configuring the LDAP server, the case of the alphabetic characters must match that on the server and on the NetScaler. If the root directory of the LDAP server is specified, all of the subdirectories are also searched to find the user attribute. In large directories, this can affect performance. For this reason, Citrix recommends that you use a specific organizational unit (OU).

The following table lists examples of user attribute fields for LDAP servers.

Table 1. User Attribute Fields for LDAP Servers

LDAP server	User attribute	Case sensitive?
Microsoft Active Directory	Server sAMAccountName	No
Novell eDirectory	cn	Yes
IBM Directory Server	uid	Yes
Lotus Domino	CN	Yes

Sun ONE directory (formerly iPlanet)	uid or cn	Yes

The following table lists examples of the base distinguished name (DN).

Table 2. Examples of Base Distinguished Name

LDAP server	Base DN
Microsoft Active Directory	DC=citrix, DC=local
Novell eDirectory	dc=citrix, dc=net
IBM Directory Server	cn=users
Lotus Domino	OU=City, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	ou=People, dc=citrix, dc=com

The following table lists examples of the bind distinguished name (DN).

Table 3. Examples of Bind Distinguished Name

LDAP server	Bind DN
Microsoft Active Directory	CN=Administrator, CN=Users, DC=citrix, DC=local
Novell eDirectory	cn=admin, dc=citrix, dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

To configure LDAP authentication by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Add**.
3. In **Name**, type a name for the policy.
4. In **Authentication Type**, select **LDAP**. Next to **Server**, click **New**.
5. In **Name**, type the name of the server.
6. Under **Server**, in **IP Address** and **Port**, type the IP address and port number of the LDAP server.
7. Under **Connection Settings**, provide the following information:

   - In **Base DN (location of users)**, type the base DN under which users are located.

     Base DN is usually derived from the Bind DN by removing the user name and specifying the group where in which are located. Examples of syntax for base DN are:

     ou=users, dc=ace, dc=com
     cn=Users, dc=ace, dc=com

   - In **Administrator Bind DN**, type the administrator bind DN for queries to the LDAP directory. Examples for syntax of bind DN are:

     domain/user name
     ou=administrator, dc=ace, dc=com
     user@domain.name (for Active Directory)
     cn=Administrator, cn=Users, dc=ace, dc=com

     For Active Directory, the group name specified as cn=groupname is required. The group name that is defined in the NetScaler must be identical to the group name that is defined on the LDAP server. For other LDAP directories, the group name either is not required or, if required, is specified as ou=groupname.

     The NetScaler binds to the LDAP server, using the administrator credentials, and then searches for the user. After locating the user, the NetScaler unbinds the administrator credentials and rebinds with the user credentials.

   - In **Administrator Password** and **Confirm Administrator Password**, type the administrator password for the LDAP server.
8. To retrieve additional LDAP settings automatically, click **Retrieve Attributes**. The fields under **Other Settings** then populate automatically. If you do not want to do this, skip to Step 12.
9. Under **Other Settings**, in **Server Logon Name Attribute**, type the attribute under which the NetScaler should look for user logon names for the LDAP server that you are

configuring. The default is `samAccountName`.

10. In **Group Attribute**, leave the default memberOf for Active Directory or change it to that of the LDAP server type you are using. This attribute enables the NetScaler to obtain the groups associated with a user during authorization.

11. In **Security Type**, select the security type. If you select **PLAINTEXT** or **TLS** for security, use port number 389. If you select SSL, use port number 636.

12. To allow users to change their LDAP password, select **Allow Password Change**. If you select **PLAINTEXT** as the security type, allowing users to change their passwords is not supported.

13. Click **Create**.

14. In the **Create Authentication Policy** dialog box, next to **Named Expressions**, select the expression, click **Add Expression**, click **Create**, and click **Close**.

After the LDAP server settings are configured on the NetScaler, bind the policy to the system global entity. For more information about binding authentication policies globally, see Binding the Authentication Policies to the System Global Entity.

Determining attributes in the LDAP directory

If you need help determining your LDAP directory attributes, you can easily look them up with the free LDAP browser from Softerra.

You can download the LDAP browser from the Softerra LDAP Administrator Web site at http://www.ldapbrowser.com. After the browser is installed, set the following attributes:

- The host name or IP address of your LDAP server.
- The port of your LDAP server. The default is 389.
- The base DN field can be left blank.
- The information provided by the LDAP browser can help you determine the base DN needed for the Authentication tab.
- The Anonymous Bind check determines whether the LDAP server requires user credentials for the the browser to connect to it. If the LDAP server requires credentials, leave the check box cleared.

After completing the settings, the LDAP browser displays the profile name in the left pane and connects to the LDAP server.

Configuring RADIUS Authentication

You can configure the NetScaler appliance to authenticate user access with one or more RADIUS servers. If you are using RSA SecurID, SafeWord, or Gemalto Protiva products, use a RADIUS server.

Your configuration might require using a network access server IP address (NAS IP) or a network access server identifier (NAS ID). When configuring your NetScaler to use a RADIUS authentication server, use the following guidelines:

- If you enable use of the NAS IP, the appliance sends its configured IP address to the RADIUS server, rather than the source IP address used in establishing the RADIUS connection.
- If you configure the NAS ID, the appliance sends the identifier to the RADIUS server. If you do not configure the NAS ID, the appliance sends its host name to the RADIUS server.
- When the NAS IP is enabled, the appliance ignores any NAS ID that was configured by using the NAS IP to communicate with the RADIUS server.

To configure RADIUS authentication by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Add**.
3. In **Name**, type a name for the policy.
4. In **Authentication Type**, select **RADIUS**.
5. Next to **Server**, click **New**.
6. In **Name**, type a name for the server.
7. Under **Server**, in **IP Address**, type the IP address of the RADIUS server.
8. In **Port**, type the port. The default is 1812.
9. Under **Details**, in **Secret Key** and **Confirm Secret Key**, type the RADIUS server secret.
10. In **NAS ID**, type the identifier number, and then click **Create**.
11. In the **Create Authentication Policy** dialog box, next to **Named Expressions**, select the expression, click **Add Expression**, click **Create**, and click **Close**.

After the RADIUS server settings are configured on the NetScaler, bind the policy to the system global entity. For more information about binding authentication policies globally, see Binding the Authentication Policies to the System Global Entity.

Choosing RADIUS authentication protocols

The NetScaler appliance supports implementations of RADIUS that are configured to use any of several protocols for user authentication, including:

- Password Authentication Protocol
- Challenge-Handshake Authentication Protocol (CHAP)
- Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP Version 1 and Version 2)

If your deployment of the NetScaler is configured to use RADIUS authentication and your RADIUS server is configured to use Password Authentication Protocol, you can strengthen user authentication by assigning a strong shared secret to the RADIUS server. Strong RADIUS shared secrets consist of random sequences of uppercase and lowercase letters, numbers, and punctuation, and are at least 22 characters long. If possible, use a random character generation program to determine RADIUS shared secrets.

To further protect RADIUS traffic, assign a different shared secret to each NetScaler appliance or virtual server. When you define clients on the RADIUS server, you can also assign a separate shared secret to each client. If you do this, you must configure separately each NetScaler policy that uses RADIUS authentication.

Shared secrets are configured on the NetScaler when a RADIUS policy is created.

Configuring IP address extraction

You can configure the NetScaler to extract the IP address from a RADIUS server. When a user authenticates with the RADIUS server, the server returns a framed IP address that is assigned to the user. The following are attributes for IP address extraction:

- Allows a remote RADIUS server to supply an IP address from the internal network for a user logged on to the NetScaler.
- Allows configuration for any RADIUS attribute using the type ipaddress, including those that are vendor encoded.

When configuring the RADIUS server for IP address extraction, you configure the vendor identifier and the attribute type.

The vendor identifier enables the RADIUS server to assign an IP address to the client from a pool of IP addresses that are configured on the RADIUS server. The vendor ID and attributes are used to make the association between the RADIUS client and the RADIUS server. The vendor ID is the attribute in the RADIUS response that provides the IP address of the internal network. A value of zero indicates that the attribute is not vendor encoded. The

attribute type is the remote IP address attribute in a RADIUS response. The minimum value is one and the maximum value is 255.

A common configuration is to extract the RADIUS attribute *framed IP address*. The vendor ID is set to zero or is not specified. The attribute type is set to eight.

To configure IP address extraction by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Open**.
3. In the **Configure Authentication Policy** dialog box, next to **Server**, click **Modify**.
4. Under **Details**, in **Group Vendor Identifier**, type the value.
5. In **Group Attribute Type**, type the value, and click **OK** twice.

Configuring TACACS+ Authentication

You can configure a TACACS+ server for authentication. Similar to RADIUS authentication, TACACS+ uses a secret key, an IP address, and the port number. The default port number is 49. To configure the NetScaler to use a TACACS+ server, provide the server IP address and the TACACS+ secret. The port needs to be specified only when the server port number in use is something other than the default port number of 49.

To configure TACACS+ authentication by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Add**.
3. In **Name**, type a name for the policy.
4. In **Authentication Type**, select **TACACS**.
5. Next to **Server**, click **New**.
6. In **Name**, type a name for the server.
7. Under **Server**, type the IP address and port number of the TACACS+ server.
8. Under **TACACS server information**, in **TACACS Key** and **Confirm TACACS key**, type the key.
9. In **Authorization**, select **ON** and click **Create**.
10. In the **Create Authentication Policy** dialog box, next to **Named Expressions**, select the expression, click **Add Expression**, click **Create**, and click **Close**.

After the TACACS+ server settings are configured on the NetScaler, bind the policy to the system global entity. For more information about binding authentication policies globally, see Binding the Authentication Policies to the System Global Entity.

Configuring NT4 Authentication

You can configure the NetScaler appliance to use Windows NT LAN Manager (NTLM) authentication to authenticate users against the user database on a Windows NT 4.0 domain controller. A Windows NT 4.0 domain controller maintains domain user accounts in a database on the Windows NT 4.0 server. A domain user account includes a user name and password and other information about the user.

When a user logs on to the NetScaler, the user enters the user name and password maintained in the domain user account on the Windows NT 4.0 server. The NetScaler connects to the Windows NT 4.0 server and passes these credentials to the server. The server authenticates the user. If you need to configure the NetScaler to authenticate clients against a Windows NT 4.0 primary or backup domain controller, you need to specify the server IP address, the domain name, and the domain administrator user name and password of the person who is authorized to administer the domain. These parameters are necessary because the NetScaler joins the domain to communicate authentication data.

NT4 authentication supports NTLMv1 and NTLMv2 authentication protocols only.

To configure NT4 authentication by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Add**.
3. In **Name**, type a name for the policy.
4. In **Authentication Type**, select **NT4**.
5. Next to **Server**, click **New**.
6. In **Server**, type the name of the server.
7. Complete the settings as they are configured on your Windows NT 4.0 server and click **Create**.
8. In the **Create Authentication Policy** dialog box, next to **Named Expressions**, select the expression, click **Add Expression**, click **Create**, and click **Close**.

When the settings for Windows NT 4.0 authentication are configured, bind the policy to the system global entity. For more information about binding authentication policies globally, see Binding the Authentication Policies to the System Global Entity.

Binding the Authentication Policies to the System Global Entity

When the authentication policies are configured, bind the policies to the system global entity.

To bind an authentication policy globally by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Global Bindings**.
3. Under **Details**, click **Insert Policy**.
4. Under **Policy Name**, select the policy and click **OK**.

To unbind a global authentication policy by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Authentication**.
2. On the **Policies** tab, click **Global Bindings**.
3. In the **Bind/Unbind Authentication Policies** dialog box, in **Policy Name**, select the policy, click **Unbind Policy** and then click **OK**.

SNMP

You can use Simple Network Management Protocol (SNMP) to configure the SNMP agent on the Citrix® NetScaler® appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the NetScaler. The traps are then sent to a remote device called a *trap listener*, which signals the abnormal condition on the NetScaler appliance. Or, you can query the SNMP agent for System-specific information from a remote device called an *SNMP manager*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The SNMP agent on the NetScaler can generate traps compliant with SNMPv1 and SNMPv2 only. For querying, the SNMP agent supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

The following figure illustrates a network with a NetScaler that has SNMP enabled and configured. In the figure, each SNMP network management application uses SNMP to communicate with the SNMP agent on the NetScaler. The SNMP agent searches its management information base (MIB) to collect the data requested by the SNMP Manager and provides the information to the application.

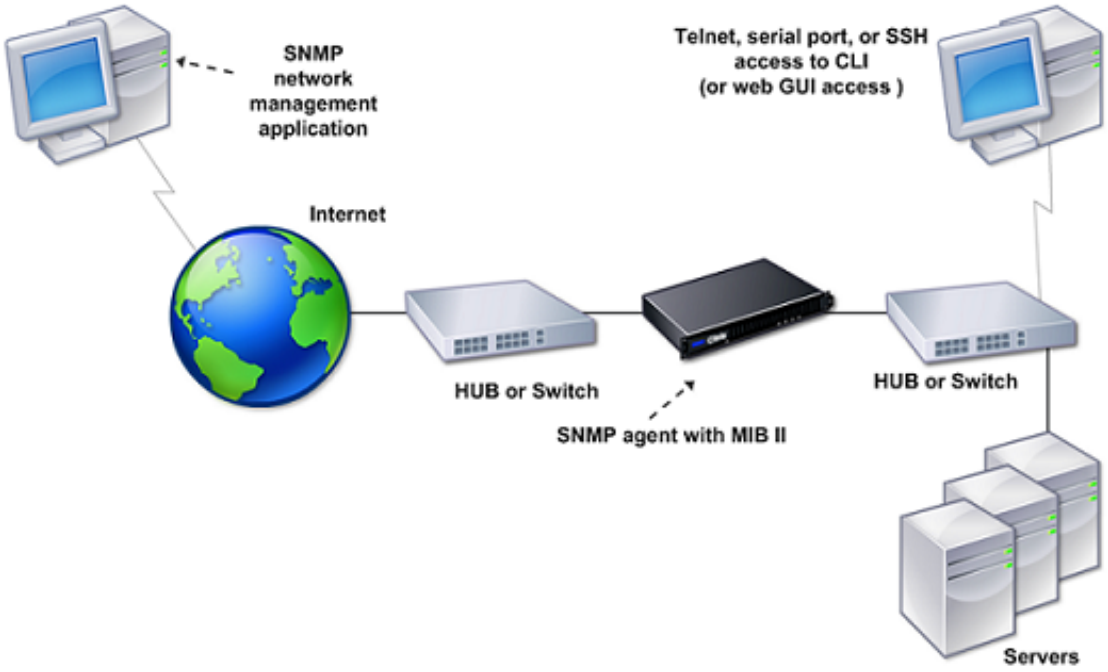


Figure 1. *NetScaler Supporting SNMP*

Importing MIB Files to the SNMP Manager and Trap Listener

You must download the following files to SNMP managers and trap listeners before you start monitoring a NetScaler appliance.

- **NS-MIB-smiv1.mib.** This file is used by SNMPv1 managers and trap listeners.
- **NS-MIB-smiv2.mib.** This file is used by SNMPv2 and SNMPv3 managers and SNMPv2 trap listeners.

The MIB files include the following:

- **A subset of standard MIB-2 groups.** Provides the MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.
- **A NetScaler enterprise MIB.** Provides NetScaler-specific configuration and statistics.

To import the MIB files to the SNMP manager and trap listener

- Logon to the **Downloads** page of NetScaler appliance GUI.
- Under **SNMP Files**, do one of the following:
  1. If your SNMP management application is other than WhatsUpGold, download the following files to your SNMP management application:
     - NS-MIB-smiv2.mib
     - NS-MIB-smiv1.mib
  2. If you are using the WhatsUpGold SNMP management application, download only the following files to the SNMP management application:
     - mib.txt
     - traps.txt

Configuring the NetScaler to Generate SNMPv1 and SNMPv2 Traps

You can configure the NetScaler to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the NetScaler. The traps are sent to a remote device called a *trap listener*. This helps administrators monitor the NetScaler and respond promptly to any issues.

The NetScaler provides a set of condition entities called *SNMP alarms*. When the condition in any SNMP alarm is met, the NetScaler generates SNMP trap messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the NetScaler appliance.

To configure the NetScaler to generate traps, you need to enable and configure alarms. Then, you specify trap listeners to which the NetScaler will send the generated trap messages.

Enabling or Disabling an SNMP Alarm

The NetScaler generates traps only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them.

When you enable an SNMP alarm, the NetScaler generates corresponding trap messages when some events occur. Some NetScaler alarms are enabled by default.

To enable or disable an SNMP alarm by using the command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- enable snmp alarm <alarm name>
- sh snmp alarm <alarm name>

Example

```
 > enable snmp alarm LOGIN-FAILURE
 Done
 > show snmp alarm LOGIN-FAILURE
 Alarm Alarm Threshold Normal Threshold Time State Severity Logging
----- --------------- ---------------- ---- -------- ------------- --------
 1) LOGIN-FAILURE N/A N/A N/A ENABLED - ENABLED
Done
```

To enable or disable an SNMP alarm by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Alarms**.
2. In the details pane, select an alarm (for example, **Login-Failure**), and do one of the following:
   - To enable an alarm, click **Enable**.
   - To disable an alarm, click **Disable**.

   A message appears in the status bar, stating that the alarm is enabled or disabled successfully.

Configuring Alarms

The NetScaler provides a set of condition entities called *SNMP alarms*. When the condition set for an SNMP alarm is met, the NetScaler generates SNMP traps messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the NetScaler appliance.

You can assign an SNMP alarm with a severity level. When you do this, the corresponding trap messages are assigned that severity level.

The following are the severity levels, defined in the NetScaler, in decreasing order of severity.

- Critical
- Major
- Minor
- Warning
- Informational

For example, if you set a Warning severity level for the SNMP alarm named LOGIN-FAILURE, the trap messages generated when there is a login failure will be assigned with the Warning severity level.

You can also configure an SNMP alarm to log the corresponding trap messages generated whenever the condition on that alarm is met.

To configure an SNMP alarm by using the command line

At the NetScaler command prompt, type the following commands to configure an SNMP alarm and verify the configuration:

- set snmp alarm <alarm Name> [**-thresholdValue** <positive_integer> [**-normalValue** <positive_integer>]] [**-time** <secs>] [**-state** (ENABLED | DISABLED)] [**-severity** <severity>] [**-logging** (ENABLED | DISABLED)]
- sh snmp alarm <alarm Name>

Parameters for configuring SNMP alarms

severity

Severity level of this alarm. Possible values: Critical, Major, Minor, Warning, Informational. Default: Informational.

logging

Enable logging of SNMP trap messages by Syslog. Possible values: ENABLED and DISABLED.

To configure SNMP alarms by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Alarms**.
2. In the details pane, select an alarm (for example, **Login-Failure**), and then click **Open**.
3. In the **Configure SNMP Alarm** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring SNMP alarms" as shown:
   - **Severity**—severity
   - **Logging**—logging
4. Click **OK**. A message appears in the status bar, stating that the alarm has been configured successfully.

Configuring Traps

After configuring the alarms, you need to specify the trap listener to which the NetScaler will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

You can also configure the NetScaler to send SNMP trap messages with a source IP, other than the NetScaler IP address (NSIP), to a particular trap listener. You can set the source IP to either a mapped IP address (MIP) or a subnet IP address (SNIP) configured on the NetScaler appliance.

You can also configure the NetScaler to send trap messages based on a severity level to a trap listener. For example, if you set the severity level as Minor for a trap listener, all

the trap messages of the severity level equal to or greater than Minor (Minor, Major, and Critical) will be sent to the trap listener.

Also, you need to specify a community string if you have defined the same in the trap listener. This string is sent in the trap messages to the specified trap listener. The trap listener will accept a trap message only when the community string defined in the trap message matches the community string defined in the trap listener; otherwise, the trap message is dropped.

To add an SNMP trap by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp trap <trapClass> <trapDestination> **-version** (V1 | V2) **-destPort** <port> **-communityName** <string> **-srcIP** <ip_addr> **-severity** <severity>
- show snmp trap

Example

```
 add snmp trap specific 10.102.29.3 -version V2 -destPort 80 -communityName com1 -severity Major
 Done
> show snmp trap
Type       DestinationIP    DestinationPort  Version     SourceIP        Min-Severity   Community
```

```
----      -------------   ---------------  -------   --------      ------------  ---------
generic    10.102.29.9     162             V2        NetScaler IP   N/A           public
specific   10.102.29.9     162             V2        NetScaler IP   -             public
specific   10.102.29.3     80              V2        NetScaler IP   Major         com1
 Done
```

Parameters for configuring SNMP traps

trapClass

The trap type. Possible values: generic and specific.

version

SNMP version of the trap PDU to be sent.

trapDestination

IP address of the trap destination.

destPort

Destination port of the trap. Default: 162. Minimum value: 1

scrIP

Source IP of the traps. Default: NetScaler IP (NSIP)

severity

Specify the severity level of trap messages. All generated trap mesages of the severity level up to the specified severity level will be sent to the trap listener.

Possible values: Critical, Major, Minor, Warning, and Informational.

Default: Informational.

communityName

The community string. Default: public.

To configure SNMP Traps by using the configuration utility

1. In the navigation pane, expand **System,** expand **SNMP,** and then click **Traps**.
2. In the details pane, do one of the following:
   - To create a new trap, click **Add**.
   - To modify an existing trap, select the trap, and then click **Open**.
3. In the **Create SNMP Trap Destination** or **Configure SNMP Trap** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring SNMP traps" as shown:
   - **Type*****—trapClass
   - **Version**—version
   - **Destination IP Address*****—trapDestination
   - **Destination Port**—destPort
   - **Source IP Address**—srcIP
   - **Minimum Severity**—severity
   - **Community Name**—communityName

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the trap has been configured successfully.

Configuring the NetScaler for SNMP v1 and v2 Queries

You can query the NetScaler SNMP agent for system-specific information from a remote device called *SNMP managers*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The following types of SNMP v1 and v2 queries are suppoted by the SNMP agent:

- GET
- GET NEXT
- ALL
- GET BULK

You can create strings called *community strings* and associate each of these to query types. You can associate one or more community strings to each query type. Community string are passwords and used to authenticate SNMP queries from SNMP managers.

For example, if you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the NetScaler appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

Specifying an SNMP Manager

You must configure the NetScaler appliance to allow the appropriate SNMP managers to query it. You must also provide the SNMP manager with the required NetScaler-specific information. You can add up to a maximum of 100 SNMP managers or networks.

You can also specify an associated host name of an SNMP manager. To do so, you need to add a DNS name server that resolves the host name of the SNMP manager to its IP address. You can add up to a maximum of five host name-based SNMP managers.

If you do not configure at least one SNMP manager, the NetScaler appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds only to SNMP queries from those specific IP addresses.

If you remove an SNMP manager from the NetScaler configuration, that manager can no longer query the NetScaler.

To add an SNMP manager by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp manager <IPAddress> ... [**-netmask** <netmask>]
- show snmp manager

Example

```
> add snmp manager 10.102.29.10
 Done
> show snmp manager
1)      10.102.29.5         255.255.255.255
Done
```

To add an SNMP manager by specifying its IP address, using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp manager <IPAddress> ... [**-netmask** <netmask>]
- show snmp manager

Example

```
> add snmp manager 10.102.29.10
 Done
> show snmp manager
1)      10.102.29.5         255.255.255.0
Done

> add snmp manager 10.102.29.15  10.102.29.30
 Done
> show snmp manager
1)      IP Address:     10.102.29.10
        Netmask:        255.255.255.255
2)      IP Address:     10.102.29.15
        Netmask:        255.255.255.255
3)      IP Address:     10.102.29.30
        Netmask:        255.255.255.255
 Done
```

To add an SNMP manager by specifying its host name, using the NetScaler command line

Important: If you specify the SNMP manager's host name instead of its IP address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see Adding a Name Server.

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp manager <IPAddress> [**-domainResolveRetry** <integer>]
- show snmp manager

Example

```
> add nameserver 10.103.128.15
Done
> show nameserver
1)      10.103.128.15  -  State: UP
```

```
 Done

> add snmp manager engwiki.eng.company.net –domainResolveRetry 10
 Done
> show snmp manager
1)      Hostname:       abc.com (Unresolved IP)
        Resolve Retry:  7
2)      Hostname:       engwiki.eng.company.net (10.217.3.249)
        Resolve Retry:  10
 Done
```

Parameters for configuring an SNMP manager

IPAddress

IP or network address of the SNMP manager. You can also specify an associated host name of an SNMP manager. If you specify a host name, you must add a DNS name server that will resolve the host name of the SNMP manager to its IP address.

netmask

Subnet of management stations. Used to grant access from entire subnets to the NetScaler appliance.

domainResolveRetry

The duration, in seconds, for which the NetScaler appliance waits to send the next DNS query to resolve the host name of the SNMP manager if the last query failed. If last query succeeds, the NetScaler waits for the TTL time. Minimum value: 5. Maximum value: 20940. Default value: 5.

To add an SNMP manager by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Managers**.
2. In the details pane, click **Add**.
3. In the **Create SNMP Manager** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP manager" as shown:
   - **IP Address***—IPAddress
   - **Netmask**—netmask

   *A required parameter
4. In the **Create SNMP Manager** dialog box, do one of the following:
   - To specify the host name of an SNMP manager, select Management Host and specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP manager" as shown:
     - **Host Name***—IPAddress
     - **Resolve Retry (secs)***—domainResolveRetry

     Important: If you specify the SNMP manager's host name instead of its IP address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see Adding a Name Server.
   - To specify the IP address of an SNMP manager, select Management Network and specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP manager" as shown:
     - **IP Address***—IPAddress
     - **Netmask**—netmask
5. Click **Create**, and then click **Close**. A message appears in the status bar, stating that the SNMP manager has been configured successfully.

Specifying an SNMP Community

You can create strings called *community strings* and associate them with the following SNMP query types on the NetScaler:

- GET
- GET NEXT
- ALL
- GET BULK

You can associate one or more community strings to each query types. For example, when you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the NetScaler appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

If you don't associate any community string to a query type then the SNMP agent responds to all SNMP queries of that type.

To specify an SNMP community by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp community <communityName> <permissions>
- sh snmp community

Example

```
> add snmp community com all
 Done
> show snmp community com
Community: com  Permissions: ALL
 Done
```

Parameters for configuring an SNMP community string

communityName

SNMP community string.

permissions

Access privileges. Possible values: GET, GET NEXT, GET BULK, ALL.

To configure an SNMP community string by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Community**.
2. In the details pane, click **Add**.
3. In the **Create SNMP Community** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP community string" as shown:
   - **Community String***—communityName
   - **Permission***—permissions

   *A required parameter
4. Click **Create**, and then click **Close**. A message appears in the status bar, stating that the SNMP community string has been configured successfully.

To remove an SNMP community string by using the configuration utility

1. In the navigation pane, expand **System**, click **SNMP**, and then click **Community**.
2. In the details pane, select the community that you want to remove (for example, **Com_All**), and then click **Remove**.

Configuring SNMP Alarms for Rate Limiting

Citrix® NetScaler® appliances such as the NetScaler MPX 10500, 12500, and 15500 are rate limited. The maximum throughput (Mbps) and packets per second (PPS) are determined by the license purchased for the appliance. For rate-limited platforms, you can configure SNMP traps to send notifications when throughput and PPS approach their limits and when they return to normal.

Throughput and PPS are monitored every seven seconds. You can configure traps with high-threshold and normal-threshold values, which are expressed as a percentage of the licensed limits. The appliance then generates a trap when throughput or PPS exceeds the high threshold, and a second trap when the monitored parameter falls to the normal threshold. In addition to sending the traps to the configured destination device, the NetScaler logs the events associated with the traps in the /var/log/ns.log file as EVENT ALERTSTARTED and EVENT ALERTENDED.

Exceeding the throughput limit can result in packet loss. You can configure SNMP alarms to report packet loss.

For more information about SNMP alarms and traps, see Configuring the NetScaler to generate SNMP v1 and v2 Traps.

Configuring an SNMP Alarm for Throughput or PPS

To monitor both throughput and PPS, you must configure separate alarms.

To configure an SNMP alarm for the throughput rate by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure the SNMP alarm and verify the configuration:

- set snmp alarm PF-RL-RATE-THRESHOLD [**-thresholdValue** <positive_integer> [**-normalValue** <positive_integer>]] [**-state** (**ENABLED** | **DISABLED**)] [**-severity** <severity>] [**-logging** (**ENABLED** | **DISABLED**)]
- show snmp alarm PF-RL-RATE-THRESHOLD

Example

```
> set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue 50
   Done

> show snmp alarm PF-RL-RATE-THRESHOLD
  Alarm                 Alarm Threshold    Normal Threshold Time State     Severity     Logging
  -----                 ---------------    ---------------- ---- --------  ------------ --------
 1) PF-RL-RATE-THRESHOLD    70                 50              N/A  DISABLED  -          ENABLED
Done
```

To modify or remove the threshold values by using the NetScaler command line

- To modify the threshold values, type the set snmp alarm PF-RL-RATE-THRESHOLD command and the parameters to be changed, with their new values.
- To remove the threshold values, type the unset snmp alarm PF-RL-RATE-THRESHOLD command, followed by the **-thresholdValue** parameter, but do not specify any value for the parameter.

Note: The normal-threshold value is automatically unset when you unset the high-threshold value.

To configure an SNMP alarm for PPS by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure the SNMP alarm for PPS and verify the configuration:

- set snmp alarm PF-RL-PPS-THRESHOLD [**-thresholdValue** <positive_integer> [**-normalValue** <positive_integer>]] [**-state** (**ENABLED** | **DISABLED**)] [**-severity** <severity>] [**-logging** (**ENABLED** | **DISABLED**)]
- show snmp alarm PF-RL-PPS-THRESHOLD

Example

```
> set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
Done

> show snmp alarm PF-RL-PPS-THRESHOLD
Alarm                                 Alarm Threshold     Normal Threshold    Time       State      S
-----                                 ---------------     ----------------    ----       -----      -
1) PF-RL-PPS-THRESHOLD 70                                 50
Done
```

To modify or remove the threshold values by using the NetScaler command line

- To modify the threshold values, type the set snmp alarm PF-RL-PPS-THRESHOLD command and the parameters to be changed, with their new values.
- To remove the threshold values, type the unset snmp alarm PF-RL-PPS-THRESHOLD command, followed by the **-thresholdValue** parameter, but do not specify any value for the parameter.

Note: The normal-threshold value is automatically unset when you unset the high-threshold value.

Parameters for configuring an SNMP alarm for throughput or PPS

thresholdValue

The high threshold value, which triggers EVENT ALERTSTARTED. Minimum value: 1.

normalValue

The normal threshold value, which triggers EVENT ALERTENDED.

state

The current state of the alarm. Possible values: ENABLED, DISABLED. Default: ENABLED.

severity

The severity level of the alarm. Possible values: Critical, Major, Minor, Warning, Informational. Default: SNMP_SEV_UNKNOWN.

logging

Log the alarm. Possible values: ENABLED, DISABLED. Default value: ENABLED.

To configure an SNMP alarm for throughput or PPS by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Alarms**.
2. In the details pane, do one of the following:
   - Select **PF-RL-RATE-THRESHOLD** to configure the SNMP alarm for throughput rate.
   - Select **PF-RL-PPS-THRESHOLD** to configure the SNMP alarm for packets per second.
3. Click **Open**.
4. In the **Configure SNMP Alarm** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring an SNMP alarm for throughput or PPS” as shown:
   - **Alarm Threshold**—thresholdValue
   - **Alarm Threshold**—thresholdValue
   - **Normal Threshold**—normalValue
   - **Severity**—severity
   - **Logging**—logging
5. Select the **Enable** check box to enable the alarm.
6. Click **OK**, and then click **Close**.

Configuring SNMP Alarm for Dropped Packets

You can configure an alarm for packets dropped as a result of exceeding the throughput limit and an alarm for packets dropped as a result of exceeding the PPS limit.

To configure an SNMP alarm for packets dropped because of excessive throughput, by using the NetScaler command line

At the NetScaler command prompt, type:

set snmp alarm PF-RL-RATE-PKTS-DROPPED [**-state** (**ENABLED** | **DISABLED**)] [**-severity** <severity>] [**-logging** (**ENABLED** | **DISABLED**)]

To configure an SNMP alarm for packets dropped because of excessive PPS, by using the NetScaler command line

At the NetScaler command prompt, type:

set snmp alarm PF-RL-PPS-PKTS-DROPPED [**-state** (**ENABLED** | **DISABLED**)] [**-severity** <severity>] [**-logging** (**ENABLED** | **DISABLED**)]

Parameters for configuring an SNMP alarm for dropped packets

state

The current state of the alarm. Possible values: ENABLED, DISABLED. Default: ENABLED.

severity

The severity level of the alarm. Possible values: Critical, Major, Minor, Warning, Informational. Default: SNMP_SEV_UNKNOWN.

logging

Log the alarm. Possible values: ENABLED, DISABLED. Default value: ENABLED.

To configure an SNMP alarm for dropped packets by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Alarms**.
2. In the details pane, do one of the following:
   - Select **PF-RL-RATE-PKTS-DROPPED** to configure an SNMP alarm for packets dropped because of excessive throughput.
   - Select **PF-RL-PPS-PKTS-DROPPED** to configure an SNMP alarm for packets dropped because of excessive PPS.
3. Click **Open**.
4. In the Configure SNMP Alarm dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring an SNMP alarm for dropped packets” as shown:
   - **Severity**—severity
   - **Logging**—logging
5. Select the **Enable** check box to enable the alarm.
6. Click **OK**, and then click **Close**.

Configuring the NetScaler for SNMPv3 Queries

Simple Network Management Protocol Version 3 (SNMPv3) is based on the basic structure and architecture of SNMPv1 and SNMPv2. However, SNMPv3 enhances the basic architecture to incorporate administration and security capabilities, such as authentication, access control, data integrity check, data origin verification, message timeliness check, and data confidentiality.

To implement message level security and access control, SNMPv3 introduces the user-based security model (USM) and the view-based access control model (VACM).

- **User-Based Security Model.** The user-based security model (USM) provides message-level security. It enables you to configure users and security parameters for the SNMP agent and the SNMP manager. USM offers the following features:
  - **Data integrity:** To protect messages from being modified during transmission through the network.
  - **Data origin verification:** To authenticate the user who sent the message request.
  - **Message timeliness:** To protect against message delays or replays.
  - **Data confidentiality:** To protect the content of messages from being disclosed to unauthorized entities or individuals.
- **View-Based Access Control Model.** The view-based access control model (VACM) enables you to configure access rights to a specific subtree of the MIB based on various parameters, such as security level, security model, user name, and view type. It enables you to configure agents to provide different levels of access to the MIB to different managers.

The Citrix NetScaler supports the following entities that enable you to implement the security features of SNMPv3:

- SNMP Engines
- SNMP Views
- SNMP Groups
- SNMP Users

These entities function together to implement the SNMPv3 security features. Views are created to allow access to subtrees of the MIB. Then, groups are created with the required security level and access to the defined views. Finally, users are created and assigned to the groups.

Note: The view, group, and user configuration are synchronized and propagated to the secondary node in a high availability (HA) pair. However, the engine ID is neither

propagated nor synchronized as it is unique to each NetScaler appliance.

To implement message authentication and access control, you need to:

- Set the Engine ID
- Configure Views
- Configure Groups
- Configure Users

Setting the Engine ID

SNMP engines are service providers that reside in the SNMP agent. They provide services such as sending, receiving, and authenticating messages. SNMP engines are uniquely identified using engine IDs.

The NetScaler has a unique engineID based on the MAC address of one of its interfaces. It is not necessary to override the engineID. However, if you want to change the engine ID, you can reset it.

To set the engine ID by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- set snmp engineId <engineID>
- show snmp engineId

Example

```
> set snmp engineId 8000173f0300c095f80c68
 Done
> show snmp engineId
       EngineID:   8000173f0300c095f80c68
 Done
```

Parameters for setting the engine ID

EngineID

Engine ID of the SNMP agent.

To set the engine ID by using configuration utility

1. In the navigation pane, expand **System,** expand **SNMP,** and then click **Users**.
2. In the details pane, click **Configure Engine ID**.
3. In the **Configure Engine ID** dialog box, in the **Engine ID** text box, type an engine ID (for example, 8000173f0300c095f80c68).
4. Click **OK**. A message appears in the status bar, stating that the engine ID has been modified successfully.

Configuring a View

SNMP views restrict user access to specific portions of the MIB. SNMP views are used to implement access control.

To add an SNMP view by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp view <name> <subtree> **-type** (included | excluded)
- sh snmp view <name>

Example

add snmp view View1 -type included

Parameters for configuring an SNMP view

name

Name of the SNMP view.

subtree

Subtree of the MIB.

type

Whether the subtree needs to be included or excluded.

To configure an SNMP view by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Views**.
2. In the details pane, click **Add**.
3. In the **Create SNMP View** or **Configure SNMP View** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP view" as shown:
   - **Name***—name
   - **Subtree***—subtree
   - **Type**—type

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the SNMP view has been configured successfully.

Configuring a Group

SNMP groups are logical aggregations of SNMP users. They are used to implement access control and to define the security levels. You can configure an SNMP group to set access rights for users assigned to that group, thereby restricting the users to specific views.

You need to configure an SNMP group to set access rights for users assigned to that group.

To add an SNMP group by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp group <name> <securityLevel> **-readViewName** <string>
- show snmp group <name> <securityLevel>

Example

```
add snmp group edocs_group2 authPriv -readViewName edocs_read_view
 Done
> show snmp group edocs_group2 authPriv
1)      Name:  edocs_group2      SecurityLevel:  authPriv
       ReadViewName:  edocs_read_view StorageType:  volatile
       Status:  active
 Done
```

Parameters for configuring an SNMP group

name

Name of the SNMP view.

securityLevel

The security level of the group. Possible values: noAuthNoPriv, authNoPriv, authPriv

readViewName

SNMP view to be associated with this group.

To configure an SNMP group by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Groups**.
2. In the details pane, click **Add**.
3. In the **Create SNMP Group** or **Configure SNMP Group** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP group" as shown:
   - **Name***—name
   - **Security Level***—securityLevel
   - **Read View Name***—readViewName

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the SNMP group has been configured successfully.

Configuring a User

SNMP users are the SNMP managers that the agents allow to access the MIBs. Each SNMP user is assigned to an SNMP group.

You need to configure users at the agent and assign each user to a group.

To configure a user by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp user <name> **-group** <string> [**-authType** (MD5 | SHA) {**-authPasswd** } [**-privType** (DES | AES) {**-privPasswd** }]]
- show snmp user <name>

Example

```
> add snmp user edocs_user -group edocs_group
 Done
> show snmp user edocs_user
1)      Name:  edocs_user       Group:  edocs_group
       EngineID:  123abc456abc788 StorageType:  volatile
       Status:  active
 Done
>
```

Parameters for configuring an SNMP user

name

The name of the SNMP user.

group

Specifyes the SNMP group name to which the SNMP user will belong.

authType

The authentication type. Possible values: MD5, SHA.

authPasswd

Enter an authentication password.

privType

The encryption type. Possible values: DES, AES.

privPasswd

The encryption password. Maximum Length: 31

To configure an SNMP user by using the configuration utility

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Users**.
2. In the details pane, click **Add**.
3. In the **Create SNMP User**or **Configure SNMP User** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SNMP user" as shown:
   - **Name***—name
   - **Group Name***—group
   - **Authentication Type**—authType
   - **Authentication Password**—authPasswd
   - **Privacy Type**—privType
   - **Privacy password**—privPasswd

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the SNMP user has been configured successfully.

Audit Logging

Auditing is a methodical examination or review of a condition or situation. The Audit Logging feature enables you to log the Citrix® NetScaler® states and status information collected by various modules in the kernel and in the user-level daemons. For audit logging, you have the options to configure SYSLOG, the native NSLOG protocol, or both.

SYSLOG is a standard protocol for logging. It has two components□ the SYSLOG auditing module, which runs on the NetScaler appliance, and the SYSLOG server, which can run on the underlying FreeBSD operating system (OS) of the NetScaler appliance or on a remote system. SYSLOG uses user data protocol (UDP) for the transfer of data.

Similarly, the native NSLOG protocol has two components□ the NSLOG auditing module, which runs on the NetScaler appliance, and the NSLOG server, which can run on the underlying FreeBSD OS of the NetScaler appliance or on a remote system. NSLOG uses transmission control protocol (TCP) for transfer of data.

When you run NSLOG or a SYSLOG server, it connects to the NetScaler appliance. The NetScaler appliance then starts sending all the log information to the SYSLOG or NSLOG server, and the server can filter the log entries before storing them in a log file. An NSLOG or SYSLOG server can receive log information from more than one NetScaler appliance and a NetScaler appliance can send log information to more than one SYSLOG server or NSLOG server.

The log information that a SYSLOG or NSLOG server collects from a NetScaler appliance is stored in a log file in the form of messages. These messages typically contain the following information:

- The IP address of a NetScaler appliance that generated the log message
- A time stamp
- The message type
- The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency)
- The message information

To configure audit logging, you first configure the audit modules on the NetScaler that involves creating audit policies and specifying the NSLOG server or SYSLOG server information. You then install and configure the SYSLOG or the NSLOG server on the underlying FreeBSD OS of the NetScaler appliance or on a remote system.

Note: Because SYSLOG is an industry standard for logging program messages and because various vendors provide support, this documentation does not include SYSLOG server configuration information.

The NSLOG server has it own configuration file (auditlog.conf). You can customize logging on the NSLOG server system by making additional modifications to the configuration file (auditlog.conf).

Configuring the NetScaler Appliance for Audit Logging

Policies define the SYSLOG or NSLOG protocol, and server actions define what logs are sent where. For server actions, you specify the system information, which runs the SYSLOG or the NSLOG server.

The Citrix NetScaler logs the following information related to TCP connections:

- Source port
- Destination port
- Source IP
- Destination IP
- Number of bytes transmitted and received
- Time period for which the connection is open

Note: You can enable TCP logging on individual load balancing vservers. You must bind the audit log policy to a specific load balancing vserver that you want to log.

Configuring Audit Servers

You can configure audit server actions for different servers and for different log levels.

To configure a SYSLOG server action by using the command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add audit syslogAction <name> <serverIP> [**-serverPort** <port>] **-logLevel** <logLevel> [**-dateFormat** (MMDDYYYY | DDMMYYYY)]
- show audit syslogAction [<name>]

Example

```
> add audit syslogaction audit-action1 10.102.1.1 -loglevel INFORMATIONAL -dateformat MMDDYYYY
 Done
> show audit syslogaction audit-action1
1)      Name: audit-action1
        Server IP: 10.102.1.1   Port: 514
        Loglevel : INFORMATIONAL
        Date Format: MMDDYYYY
        Time Zone: GMT_TIME
        Facility: LOCAL0
        Tcp Logging: NONE
        ACL Logging: DISABLED
        UserDefinedLogging: No
           AppFlow export: DISABLED
 Done
```

To configure an NSLOG server action by using the command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add audit nslogAction <name> <serverIP> [**-serverPort** <port>] **-logLevel** <logLevel> [**-dateFormat** (MMDDYYYY | DDMMYYYY)]
- show audit nslogAction [<name>]

Example

```
 > add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -loglevel INFORMATIONAL -dateFormat MMI
 Done
> show nslogAction nslog-action1
1)      Name: nslog-action1
        Server IP: 10.102.1.3   Port: 520
        Loglevel : INFORMATIONAL
        Date Format: MMDDYYYY
        Time Zone: GMT_TIME
        Facility: LOCAL0
        Tcp Logging: NONE
        ACL Logging: DISABLED
               UserDefinedLogging: No
           AppFlow export: DISABLED
 Done
```

Parameters for configuring auditing servers

name

The name of the SYSLOG server action or NSLOG server action.

serverIP

IP address of the auditing server.

serverPort

Port through which to communicate.

logLevel

Severity levels of messages to be logged. Possible values: ALL, NONE, or one or more of the following:

- EMERGENCY
- ALERT
- CRITICAL

- ERROR
- WARNING
- NOTICE
- INFORMATION
- DEBUG

dateFormat

Format of the date stamp. Possible values: MMDDYYYY, DDMMYYYY.

logFacility

The Facility value (RFC 3164) assigned to the log message. Uses numerical codes 0 to 7 to indicate the type of message originating from the NetScaler (for example, NS and VPN). Possible values: LOCAL0 to LOCAL7. Default: LOCAL0.

timeZone

Time zone for the time stamp. Possible values: GMT and Local. Default: Local.

tcp

Log TCP events. Possible values: NONE, ALL.

acl

Log ACL events. Possible values: ENABLED, DISABLED.

userDefinedAuditlog

Enable user-configurable log messages. Possible values: YES, NO.

appflowExport

Export log messages to the AppFlow collectors. Possible values: ENABLED, DISABLED. Default: DISABLED.

Log levels defined

EMERGENCY

Log errors indicating that the NetScaler is experiencing a critical problem that may make it unusable.

ALERT

Log problems that are not critical to current operations but that indicate a need for immediate corrective action to prevent a critical problem.

CRITICAL

Log critical conditions, which do not restrict current operations but may escalate to a larger problem.

ERROR

Log messages related to failed NetScaler operations.

WARNING

Log issues that may result in critical errors.

NOTICE

Log events specified by the INFORMATION setting, but in greater detail.

INFORMATION

Log actions taken by the NetScaler. This level is useful for troubleshooting problems.

DEBUG

Log extensive, detailed information to help developers troubleshoot problems.

To configure an auditing server action

1. In the navigation pane, expand **System**, expand **Auditing**, and then click **Policies**.
2. In the details pane, on the **Servers** tab, do one of the following:
   - To create a new server action, click **Add**.
   - To modify an existing server action, select the server, and then click **Open**.
3. In the **Create Auditing Server** or **Configure Auditing Server** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring auditing servers" as shown:
   - **Name***—name
   - **IP Address***—serverIP
   - **Port**—serverPort
   - **Log Levels**—logLevel
   - **Log Facility**—logFacility
   - **Date format**—dateFormat
   - **Time Zone**—timeZone
   - **TCP Logging**—tcp
   - **ACL Logging**—acl
   - **User Configurable Log Messages**—userDefinedAuditlog
   - **AppFlow Export**—appflowExport

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the auditing server has been configured successfully.

Configuring Audit Policies

The audit policies define the SYSLOG or NSLOG protocol.

To configure a SYSLOG policy by using the command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add audit syslogPolicy <name> <rule> <action>
- show audit syslogPolicy [<name>]

Example

```
> add audit syslogpolicy syslog-pol1 ns_true audit-action1
Done
> show audit syslogpolicy syslog-pol1
1)      Name: syslog-pol1       Rule: ns_true
       Action: audit-action1
 Done
```

To configure an NSLOG policy by using the command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add audit nslogPolicy <name> <rule> <action>
- show audit nslogPolicy [<name>]

Example

```
> add audit nslogPolicy nslog-pol1 ns_true nslog-action1
 Done
> show audit nslogPolicy nslog-pol1
1)      Name: nslog-pol1        Rule: ns_true
        Action: nslog-action1
 Done
```

Parameters for configuring audit policies

name

The name of NSLOG policy or SYSLOG policy.

rule

The name of the rule or expression that the policy will use. It currently supports only the rule "ns_true."

This parameter is only for the command line.

In the configuration utility ns_true is internally assigned as a rule for the SYSLOG or the NSLOG policy.

action

SYSLOG server action or the NSLOG server action. NSLOG server action is bind to a NSLOG audit policy and SYSLOG server action is bind to a SYSLOG audit policy.

To configure an audit server policy

1. In the navigation pane, expand **System**, expand **Auditing**, and then click **Policies**.
2. In the details pane, on the **Policies** tab, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Open**.
3. In the **Create Auditing Policy** or **Configure Auditing Policy** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring auditing policies" as shown:
   - **Name*** ▫ name
   - **Server*** ▫ action

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the auditing policy has been configured successfully.

Binding the Audit Policies Globally

You must globally bind the audit log policies to enable logging of all Citrix® NetScaler® system events. By defining the priority level, you can set the evaluation order of the audit server logging. Priority 0 is the highest and is evaluated first. The higher the priority number, the lower is the priority of evaluation.

To configure a SYSLOG policy by using the command line

- bind system global [<policyName> [**-priority** <positive_integer>]]
- sh system global

Example

```
> bind system global nslog-pol1 -priority 20
  Done

> sh system global
1)      Policy Name: nslog-pol1 Priority: 20
2)      Policy Name: syslog-pol1 Priority: 50
3)      Policy Name: nslogpol9   Priority: 100
 Done
```

Parameters for binding the audit policies globally

policyName

The name of the NSLOG or SYSLOG policy.

priority

A numeric value that indicates when this policy is evaluated relative to others. A lower priority is evaluated before a higher one.

To globally bind the audit policy

1. In the navigation pane, expand **System**, expand **Auditing**, and then click **Policies**.
2. In the details pane, on the **Policies** tab, click **Global Bindings**.
3. In the **Bind/Unbind Auditing Global Policies** dialog box, click **Insert Policy**.
4. Select the policy from the drop-down list that appears under **Policy Name**, and then click **OK**. A message appears in the status bar, stating that the auditing policy has been globally bound.

Configuring Policy-Based Logging

You can configure policy-based logging for rewrite and responder policies. Audit messages are then logged in a defined format when the rule in a policy evaluates to TRUE. To configure policy-based logging, you configure an audit-message action that uses default syntax expressions to specify the format of the audit messages, and associate the action with a policy. The policy can be bound either globally or to a load balancing or content switching virtual server. You can use audit-message actions to log messages at various log levels, either in syslog format only or in both syslog and newnslog formats.

Pre Requisites

User Configurable Log Messages (userDefinedAuditlog) option is enabled for when configuring the audit action server to which you want to send the logs in a defined format. For more information about enabling policy-based logging on a audit action server, see http://support.citrix.com/proddocs/topic/netscaler-admin-guide-93/ns-ag-bind-adt-pol-tsk.html.

- The related audit policy is bound to system global. For more information about binding audit policies to system global, see http://support.citrix.com/proddocs/topic/netscaler-admin-guide-93/ns-ag-bind-adt-pol-tsk.html.

Configuring an Audit Message Action

You can configure audit message actions to log messages at various log levels, either in syslog format only or in both syslog and newnslog formats. Audit-message actions use expressions to specify the format of the audit messages.

To create an audit message action by using the NetScaler command line

At the NetScaler command prompt, type:

add audit messageaction <name> <logLevel> <stringBuilderExpr> [-**logtoNewnslog** (YES|NO)] [-**bypassSafetyCheck** (YES|NO)]

To modify or remove an audit message action by using the NetScaler command line

- To modify an audit message action, type the set audit messageaction command, the name of the action, and the parameters to be changed, with their new values.
- To remove an audit message action, type the rm audit messageaction command and the name of the action.

Example

```
> add audit messageaction log-act1 CRITICAL "'Client:"+CLIENT.IP.SRC+" accessed "+H
TTP.REQ.URL' -bypassSafetyCheck YES
 Done

> show audit messageaction log-act1

1)     Name: log-act1
       LogMsgStr: "Client:"+CLIENT.IP.SRC+" accessed "+HTTP.REQ.URL
       Loglevel:CRITICAL
       Log2Newnslog:NO
       BypassSafetyCheck : YES
       Hits: 0
       Undef Hits: 0
       Action Reference Count: 0
 Done
```

Parameters for configuring an audit message action

name

The name of the audit message action. The name can begin with a letter, number, or the underscore symbol, and can consist of up to 127 characters including letters, numbers, and hyphen (-), period (.) pound (#), space ( ), at sign (@), equal sign (=), colon (:), and underscore (_) symbols.

logLevel

The log level for the message action. Possible values: EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE.

stringBuilderExpr

The expression that defines the format of the log message. For a complete description of NetScaler expressions, see Policy Configuration and Reference.

bypassSafetyCheck

Bypass the safety check and allow unsafe expressions. Possible values: YES, NO. Default: NO.

logtoNewnslog

Log messages in newnslog format in addition to logging them in syslog format. Possible values: YES, NO. Default: NO.

To configure an audit message action by using the configuration utility

1. In the navigation pane, expand **System**, expand **Auditing**, and then click **Message Actions**.
2. In the details pane, do one of the following:
   - To create a new audit message action, click **Add**.
   - To modify an existing audit message action, select the action, and then click **Open**.
3. In the **Create Message Action** or **Configure Message Action** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an audit message action" as shown:
   - **Name***—name
   - **Log Level***—logLevel
   - **Log Message**—stringBuilderExpr
   - **Bypass Safety Check**—bypassSafetyCheck (To specify **YES**, select the check box.)
   - **Log in newnslog**—logtoNewnslog (To specify **YES**, select the check box.)

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. The audit message action that you configured appears in the details pane.

Binding Audit Message Action to a Policy

After you have created an audit message action, you must bind it to a rewrite or responder policy. For more information about binding log message actions to a rewrite or responder policy, see Rewrite or Responder.

Installing and Configuring the NSLOG Server

During installation, the NSLOG server executable file (auditserver) is installed along with other files. The auditserver executable file includes options for performing several actions on the NSLOG server, including running and stopping the NSLOG server. In addition, you use the auditserver executable to configure the NSLOG server with the IP addresses of the NetScaler appliances from which the NSLOG server will start collecting logs. Configuration settings are applied in the NSLOG server configuration file (auditlog.conf).

Then, you start the NSLOG server by executing the auditserver executable. The NSLOG server configuration is based on the settings in the configuration file. You can further customize logging on the NSLOG server system by making additional modifications to the NSLOG server configuration file (auditlog.conf).

The following table lists the operating systems on which the NSLOG server is supported.

Table 1. Supported Platforms for the NSLOG Server

Operating system	Software requirements
Windows	• Windows XP Professional • Windows Server 2003 • Windows 2000/NT
Linux	• Red Hat Enterprise Linux AS release 4 (Nahant) - Linux version 2.6.9-5.EL • Red Hat 3.4.3-9.EL4 - Linux version 2.6.9-5.ELsmp • Red Hat Linux 3.2.2-5 - Linux version 2.4.20-8
FreeBSD	FreeBSD 4.9

The minimum hardware specifications for the platform running the NSLOG server are as follows:

- Processor- Intel x86 ~501 megahertz (MHz)
- RAM - 512 megabytes (MB)
- Controller - SCSI

Installing NSLOG Server on the Linux Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the Linux system as an adminstrator. Use the following procedure to install the NSLOG server executable files on the system.

To install the NSLOG server package on a Linux operating system

1. At a Linux command prompt, type the following command to copy the NSauditserver.rpm file to a temporary directory:

   cp <path_to_cd>**/Utilities/auditserver/Linux/NSauditserver.rpm /tmp**

2. Type the following command to install the NSauditserver.rpm file:

   rpm **-i NSauditserver.rpm**

   This command extracts the files and installs them in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

To uninstall the NSLOG server package on a Linux operating system

1. At a command prompt, type the following command to uninstall the audit server logging feature:

   rpm **-e NSauditserver**

2. For more information about the NSauditserver RPM file, use the following command:

   rpm **-qpi *.rpm**

3. To view the installed audit server files use the following command:

   rpm **-qpl *.rpm**

   *.rpm: Specifies the file name.

Installing NSLOG Server on the FreeBSD Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the FreeBSD operating system as an adminstrator. Use the following procedure to install the NSLOG server executable files on the system.

Note: If you want to install the NSLOG server on the underlying FreeBSD OS of the NetScaler appliance, then log on to the NetScaler appliance's command line and then switch to the shell prompt and then use the following procedure.

To install the NSLOG server package on a FreeBSD operating system

1. Copy the NSauditserver.tgz file to a target directory by using the following command:

   cp <path_to_cd>**/Utilities/auditserver/Freebsd/NSauditserver.tgz /<targetdirectory>**

2. Change to the target directory:

   cd / <targetdirectory>

3. Use the following command to install the package:

   pkg_add NSauditserver.tgz

   This command extracts the files and installs them in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

4. At a command prompt, type the following command to check whether the package is installed:

   pkg_info | grep NSauditserver

To uninstall the NSLOG server package on a FreeBSD operating system

At a command prompt, type:

```
pkg_delete NSauditserver
```

Installing NSLOG Server Files on the Windows Operating System

You need to copy the NSLOG package from the NetScaler product CD or download it from www.citrix.com. The NSLOG package has the following name format AuditServer _<release number>-<build number>.zip (for example, AuditServer_9.3-51.5.zip). This package contains NSLOG installation packages for all the supported platforms.

To download NSLOG package from www.Citrix.com

1. From any system, open www.citrix.com in the Web browser.
2. In the menu bar, click **Log In**.
3. Enter your login credentials and then click **Log In**.
4. In the menu bar, click **Downloads**.
5. Search to the page of the desired release number and build.
6. On the desired page, under **Audit Servers**, click **Download** to download a file, having the format `AuditServer_<release number>-<build number>.zip` , to your local system (for example, `AuditServer_9.3-51.5.zip`).
7. Extract `audserver_<release number>-<build number>.zip` (for example, `audserver_win-9.3-51.5.zip`) from the AuditServer_<release number>-<build number>.zip (for example, `AuditServer_9.3-51.5.zip`).
8. Copy the extracted file `audserver_<release number>-<build number>.zip` (for example, `audserver_win-9.3-51.5.zip`) to a Windows system on which you want to install the NSLOG server.

To install NSLOG server on a Windows operating system

1. Log on to the Windows system as an administrator.
2. Unzip the `audserver_<release number>-<build number>.zip` file (for example, `audserver_win-9.3-51.5.zip`).
3. The following directories are extracted:
   a. <root directory extracted from the Windows NSLOG server package zip file>\bin (for example, `C:\audserver_win-9.3-51.5\bin`)
   b. <root directory extracted from the Windows NSLOG server package zip file>\etc (for example, `C:\audserver_win-9.3-51.5\ etc`)
   c. < root directory extracted from the Windows NSLOG server package zip file >\samples (for example, `C:\audserver_win-9.3-51.5\ samples`)
4. At a command prompt, run the following command from the <root directory extracted from the Windows `NSLOG server package zip file>\bin path:`

   audserver **-install -f** <directorypath>**auditlog.conf**

   <directorypath>: Specifies the path to the configuration file (`auditlog.conf`). By default, log.conf is under <root directory extracted from Windows `NSLOG server package zip file>\samples` directory. But you can copy auditlog.conf to your desired directory.

To uninstall the NSLOG server on a Windows operating system

At a command prompt, run the following from the `<root directory extracted from Windows NSLOG server package zip file>\bin path:`

audserver **-remove**

NSLOG Server Command Options

The following table describes the commands that you can use to configure audit server options.

Table 1. Audit Server Options

Audit server commands	Specifies
audserver **-help**	The available Audit Server options.
audserver **-addns -f** <path to configuration file>	The system that gathers the log transaction data. You are prompted to enter the IP address of the NetScaler appliance. Enter the valid user name and password.
audserver **-verify -f** <path to configuration file>	Check for syntax or semantic errors in the configuration file (for example, auditlog.conf).
audserver **-start -f** <path to configuration file>	Start audit server logging based on the settings in the configuration file (auditlog.conf) Linux only: To start the audit server as a background process, type the ampersand sign (&) at the end of the command.
audserver **-stop** (Linux only)	Stops audit server logging when audit server is started as a background process. Alternatively, use the Ctrl+C key to stop audit server logging.
audserver **-install -f** <path to configuration file> (Windows only)	Installs the audit server logging client as a service on Windows.

audserver **-startservice** (Windows Only)	Start the audit server logging service, when you enter this command at a command prompt. You can also start audit server logging from **Start > Control Panel > Services**. **Note:** Audit server logging starts by using the configuration settings in the configuration file, for example, auditlog.conf file specified in the audit server install option.
audserver **-stopservice** (Windows Only)	Stop audit server logging.
audserver **-remove**	Removes the audit server logging service from the registry.

Run the audserver command from the directory in which the audit server executable is present:

- On Windows: \ns\bin
- On Solaris and Linux: \usr\local\netscaler\bin

The audit server configuration files are present in the following directories:

- On Windows: \ns\etc
- On Linux: \usr\local\netscaler\etc

The audit server executable is started as ./auditserver in Linux and FreeBSD.

Adding the NetScaler Appliance IP Addresses on the NSLOG Server

In the configuration file (auditlog.conf), add the IP addresses of the NetScaler appliances whose events must be logged.

To add the IP addresses of the NetScaler appliance

At a command prompt, type the following command:

audserver **-addns -f** <directorypath>**auditlog.conf**

<directorypath>: Specifies the path to the configuration file (auditlog.conf).

You are prompted to enter the information for the following parameters:

NSIP: Specifies the IP address of the NetScaler appliance, for example, 10.102.29.1.

Userid: Specifies the user name, for example, nsroot.

Password: Specifies the password, for example, nsroot.

If you add multiple NetScaler IP addresses (NSIP), and later you do not want to log all of the NetScaler appliance event details, you can delete the NSIPs manually by removing the NSIP statement at the end of the auditlog.conf file. For a high availability (HA) setup, you must add both primary and secondary NetScaler IP addresses to auditlog.conf by using the audserver command. Before adding the IP address, make sure the user name and password exist on the system.

Verifying the NSLOG Server Configuration File

Check the configuration file (audit log.conf) for syntax correctness to enable logging to start and function correctly.

To verify configuration, at a command prompt, type the following command:

audserver **-verify -f** <directorypath>**auditlog.conf**

<directorypath>: Specifies the path to the configuration file (audit log.conf).

Running the NSLOG Server

To start audit server logging

Type the following command at a command prompt:

audserver **-start -f** <directorypath>**auditlog.conf**

<directorypath>: Specifies the path to the configuration file (audit log.conf).

To stop audit server logging that starts as a background process in FreeBSD or Linux

Type the following command:

audserver **-stop**

To stop audit server logging that starts as a service in Windows

Type the following command:

audserver **-stopservice**

Customizing Logging on the NSLOG Server

You can customize logging on the NSLOG server by making additional modifications to the NSLOG server configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the server system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information from a NetScaler appliance or a set of NetScaler appliances.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

Creating Filters

You can use the default filter definition located in the configuration file (audit log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

Note: For consolidated logging, if a log transaction occurs for which there is no filter definition, the default filter is used (if it is enabled.) The only way you can configure consolidated logging of all the Citrix NetScaler appliances is by defining the default filter.

To create a filter

At the command prompt, type the following command in the configuration file (auditlog.conf) :

filter <filterName> **[IP** <ip>] [**NETMASK** <mask>] [ON | OFF]

<filterName>: Specify the name of the filter (maximum of 64 alphanumeric characters).

<ip>: Specify the IP addresses.

<mask>: Specify the subnet mask to be used on a subnet.

Specify ON to enable the filter to log transactions, or specify OFF to disable the filter. If no argument is specified, the filter is ON

Examples

```
filter F1 IP 192.168.100.151 ON
```

To apply the filter F2 to IP addresses 192.250.100.1 to 192.250.100.254:

```
filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON
```

filterName is a required parameter if you are defining a filter with other optional parameters, such as IP address, or the combination of IP address and Netmask.

Specifying Log Properties

Log properties associated with the filter are applied to all the log entries present in the filter. The log property definition starts with the key word BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
   logFilenameFormat ...
   logDirectory ...
   logInterval ...
   logFileSizeLimit ....
END
```

Entries in the definition can include the following:

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:
  - Static: A constant string that specifies the absolute path and the file name.
  - Dynamic: An expression that includes the following format specifiers:
    - Date (%{format}t)
    - % creates file name with NSIP

  Example

  LogFileNameFormat Ex%{%m%d%y}t.log

  This creates the first file name as Exmmddyy.log. New files are named: Exmmddyy.log.0, Exmmddyy.log.1, and so on. In the following example, the new files are crated when the file size reaches 100MB.

  Example

  ```
  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%{%m%d%y}t
  ```

  Caution: The date format %t specified in the LogFilenameFormat parameter overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat parameter.
- **logDirectory** specifies the directory name format of the log file. The name of the file can be either of the following:

- Static: Is a constant string that specifies the absolute path and file name.
- Dynamic: Is an expression containing the following format specifiers:
  - Date (%{format}t)
  - % creates directory with NSIP

The directory separator depends on the operating system. In Windows, use the directory separator \.

Example:

LogDirectory dir1\dir2\dir3

In the other operating systems (Linux, FreeBsd, Mac, etc.), use the directory separator /.

Example:

LogDirectory dir1/dir2/dir3

In the other operating systems (Linux, FreeBsd, Mac, etc.), use the directory separator /.

Example:

LogDirectory dir1/dir2/dir3

- **LogInterval** specifies the interval at which new log files are created. Use one of the following values:
  - Hourly: A file is created every hour. Default value.
  - Daily: A file is created very day at midnight.
  - Weekly: A file is created every Sunday at midnight.
  - Monthly : A file is created on the first day of the month at midnight.
  - None: A file is created only once, when audit server logging starts.
  - Size: A file is created only when the log file size limit is reached.

Example

LogInterval Hourly

LogFileSizeLimit specifies the maximum size (in MB) of the log file. A new file is created when the limit is reached.

Note that you can override the loginterval property by assigning size as its value.

The default LogFileSizeLimit is 10 MB.

Example

LogFileSizeLimit 35

Default Settings for the Log Properties

The following is an example of the default filter with default settings for the log properties:

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
 logFilenameFormat  auditlog%{%y%m%d}t.log
 end default
```

Following are two examples of defining the default filters:

Example 1

```
Filter f1 IP 192.168.10.1
```

This creates a log file for NSI 192.168.10.1 with the default values of the log in effect.

Example 2

```
Filter f1 IP 192.168.10.1
begin f1
   logFilenameFormat logfiles.log
end f1
```

This creates a log file for NSIP 192.168.10.1. Since the log file name format is specified, the default values of the other log properties are in effect.

Sample Configuration File (audit.conf)

Following is a sample configuration file:

```
################################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
################################
MYIP <NSAuditserverIP>
MYPORT 3023
#   Filter filter_nsip  IP <Specify the NetScaler IP address to filter on > ON
#   begin filter_nsip
#       logInterval         Hourly
#       logFileSizeLimit    10
#       logDirectory        logdir\%A\
#       logFilenameFormat   nsip%{%d%m%Y}t.log
#   end filter_nsip
Filter default
begin default
   logInterval      Hourly
   logFileSizeLimit    10
   logFilenameFormat   auditlog%{%y%m%d}t.log
end default
```

Web Server Logging

You can use the Web server logging feature to send logs of HTTP and HTTPS requests to a client system for storage and retrieval. This feature has two components: the Web log server, which runs on the Citrix® NetScaler® appliance, and the NetScaler Web Logging (NSWL) client, which runs on the client system. When you run the client, it connects to the NetScaler. The NetScaler buffers the HTTP and HTTPS request log entries before sending them to the NSWL client, and the client can filter the entries before storing them. You can log HTTP and HTTPS requests for all of your Web servers on one NSWL client system.

To configure Web server logging, you first enable the Web logging feature on the NetScaler and configure the size of the buffer for temporarily storing the log entries. Then, you install NSWL on the client system. You then add the NetScaler IP address (NSIP) to the NSWL configuration file. You are now ready to start the NSWL client to begin logging. You can customize Web server logging by making additional modifications to the NSWL configuration file (log.conf).

Configuring the NetScaler Appliance for Web Server Logging

On the NetScaler appliance you need to enable the Web logging feature, and you can modify the size of the buffer that stores the logged information before sending the logged information to the NetScaler Web Logging (NSWL) client.

Enabling or Disabling Web Server Logging

Web server logging is enabled by default.

To enable or disable Web server logging by using the NetScaler command line

At the NetScaler command prompt, type the following relevant commands to add or remove Web server logging and verify the configuration:

- enable ns feature WL
- disable ns feature WL
- sh ns feature

Example

```
> enable ns feature WL
Done
sh ns feature

      Feature           Acronym    Status
      -------           -------    ------
 1)   Web Logging       WL         ON
 2)   Surge Protection  SP         ON
 .
 .
 .
 24)  NetScaler Push     push       OFF
 Done
>
> disable ns feature WL
Done
sh ns feature

      Feature           Acronym    Status
      -------           -------    ------
 1)   Web Logging       WL         OFF
 2)   Surge Protection  SP         ON
 .
 .
```

.
24) NetScaler Push push OFF
Done

To enable or disable Web server logging by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, under **Modes and Features**, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **Web Logging** check box to enable the Web logging feature, or clear the check box to disable the feature.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the feature has been enabled or disabled.

Modifying the Default Buffer Size

You can change the default buffer size of 16 megabytes (MB) for Web server logging to suit your requirements. To activate your modification, you must disable and reenable Web server logging.

To modify the buffer size by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify the buffer size and and verify the configuration:

- set weblogparam-**bufferSizeMB** <size>
- sh weblogparam

Example

```
> set weblogparam -bufferSizeMB 32

> sh weblogparam
       Web Logging parameters:
        Log buffer size: 32MB
 Done
```

Parameter for modifying the buffer size

Buffer Size

Memory (in megabytes) allocated for buffering the HTTP and HTTPS request log entries before sending them to the NSWL client.

To modify the buffer size by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change global system settings**.
3. In the **Configure Global Settings** dialog box, under **Web Logging**, enter a value in the **Buffer_Size (in MBytes)** text box (for example, **32**).
4. Click **OK**.

Installing and Configuring the Client System for Web Server Logging

During installation, the NSWL client executable file (nswl) is installed along with other files. The nswl executable file includes options for performing several actions on the NSWL client, including running and stopping the NSWL client. In addition, you use the nswl executable to configure the NSWL client with the IP addresses of the NetScaler appliances from which the NSWL client will start collecting logs. Configuration settings are applied in the NSWL client configuration file (log.conf).

Then, you start the NSWL client by executing the nswl executable. The NSWL client configuration is based on the settings in the configuration file. You can further customize logging on the NSWL client system by making additional modifications to the NSLOG server configuration file (auditlog.conf).

The following table lists the operating systems on which the NSWL client is supported.

Table 1. Supported Platforms for the NSWL Client

Operating system	Version
Windows	• Windows XP Professional • Windows Server 2003 • Windows 2000/NT • Windows Server 2008 • Windows Server 2008 R2
Mac OS	Mac OS 8.6 or later
Linux	• RedHat Linux 4 or later • SUSE Linux Enterprise 9.3 or later
Solaris	Solaris Sun OS 5.6 or later
FreeBSD	FreeBSD 6.3 or later

The following table describes the minimum hardware specifications for the platform running the NSWL client.

Table 2. Minimum Hardware Specification for Platforms Running the NSWL Client

Operating system	Hardware requirements

For Windows / Linux / FreeBSD	• Processor- Intel x86 ~501 megahertz (MHz) • RAM - 512 megabytes (MB) • Controller - SCSI
For Solaris 2.6	• Processor - UltraSPARC-IIi 400 MHz • RAM - 512 MB • Controller - SCSI

If the NSWL client system cannot process the log transaction because of a CPU limitation, the Web log buffer overruns and the logging process reinitiates.

Caution: Reinitiation of logging can result in loss of log transactions.

To temporarily solve a NSWL client system bottleneck caused by a CPU limitation, you can tune the Web server logging buffer size on the NetScaler appliance. To solve the problem, you need a client system that can handle the site's throughput.

Installing NSWL Client on a Solaris Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the Solaris system as an adminstrator. Use the following procedure to install the NSWL executable and the other files on the system.

To install the NSWL client package on a Solaris operating system

1. At a command prompt, copy the NSweblog.tar file into a temporary directory using the command:

   cp <path_to_cd>/Utilities/weblog/Solaris/NSweblog.tar /tmp

2. Change to the temporary directory:

   cd /tmp

3. Extract the files from the *.tar file with the following command:

   tar xvf NSweblog.tar

   A directory NSweblog is created in the temporary directory, and the files are extracted to the NSweblog directory.

4. Install the package with the following command:

   pkgadd -d

   The list of available packages appears. In the following example, one NSweblog package is shown:

   ```
    1 NSweblog NetScaler Weblogging
   (SunOS,sparc) 7.0
   ```

5. You are prompted to select the packages. Select the package number of the NSweblog to be installed.

   After you select the package number and press Enter, the files are extracted and installed in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

6. At a command prompt, type the following command to check whether the package is installed:

   pkginfo | grep NSweblog

To uninstall the NSWL client package on a Solaris operating system

At a command prompt, type:

pkgrm NSweblog

Installing NSWL Client on a Linux Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the Linux system as an administrator. Use the following procedure to install the NSWL executable and the other files on the system.

To install the NSWL client package on a Linux operating system

1. At a command prompt, copy the NSweblog.rpm file into a temporary directory:

   cp <path_to_cd>/Utilities/weblog/Linux/NSweblog.rpm /tmp

2. To install the NSWL executable, use the following command:

   rpm -i NSweblog.rpm

   This command extracts the files and installs them in the following directories.

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

To uninstall the NSWL client package on a Linux operating system

At a command prompt, type:

rpm -e NSweblog

To get more information about the NSweblog RPM file

At a command prompt, type:

rpm -qpi *.rpm

To view the installed Web server logging files

At a command prompt, type:

rpm -qpl *.rpm

Installing NSWL Client on a FreeBSD Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the FreeBSD system as an administrator. Use the following procedure to install the NSWL executable and the other files on the system.

To install the NSWL client package on a FreeBSD operating system

1. At a command prompt, copy the NSweblog.tgz file into a temporary directory:

   cp <path_to_cd>/Utilities/weblog/Freebsd/NSweblog.tgz /tmp

2. Change to the temporary directory:

   cd /tmp

3. Install the package using the following command:

   pkg_add NSweblog.tgz

   This command extracts the files and installs them in the following directories.

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

4. To verify that the package is installed, use the following command:

   pkg_info | grep NSweblog

To uninstall the NSWL client package on a FreeBSD operating system

At a command prompt, type:

pkg_delete NSweblog

Installing NSWL Client on a Mac OS Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the Mac OS operating system as an administrator. Use the following procedure to install the NSWL executable and the other files on the system.

To install the NSWL client package on a Mac OS operating system

1. At a command prompt, copy the NSweblog.tgz file into a temporary directory with the following command:

   cp <path_to_cd>/Utilities/weblog/macos/NSweblog.tgz /tmp

2. Change to the temporary directory:

   cd /tmp

3. To install the package, use the pkg_add command:

   pkg_add NSweblog.tgz

   This command extracts the files and installs them in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

4. To verify that the package is installed, use the following command:

   pkg_info | grep NSweblog

To uninstall the NSWL client package on a Mac OS operating system

At a command prompt, type:

pkg_delete NSweblog

Installing NSWL Client on a Windows Operating System

Before installing the NSWL client, you have to copy the NSWL client package from the NetScaler product CD or download it from www.citrix.com. The NSWL client package has the following name format:

Weblog_<release number>-<build number>.zip (for example, Weblog_9.3-51.5.zip). Within the package are separate installation packages for each supported platforms.

To download NSWL client package from www.Citrix.com

1. From any system, open www.citrix.com in the Web browser.
2. In the menu bar, click **Log In**.
3. Enter your login credentials and then click **Log In**.
4. In the menu bar, click **Downloads** .
5. Search to the page of the desired release number and build.
6. On the desired page, under **Weblog Clients**, click **Download** to download a file, having the format Weblog_<release number>-<build number>.zip , to your local system (for example, Weblog_9.3-51.5.zip).
7. Extract `nswl_<release number>-<build number>.zip` (for example, `nswl_win-9.3-51.5.zip`) from the `Weblog_<release number>-<build number>.zip` (for example, `Weblog_9.3-51.5.zip`).
8. Copy the extracted file `nswl_<release number>-<build number>.zip` (for example, `nswl_win-9.3-51.5.zip`) to a Windows system on which you want to install the NSWL client.

To install the NSWL client on a Windows system

1. Log on to the Windows system as an administrator.

2. Unzip the `nswl_<release number>-<build number>.zip` file (for example , `nswl_win-9.3-51.5.zip`). The following directories are extracted:

   a. `<root directory extracted from the Windows NSWL client package zip file>\bin` (for example, `C:\nswl_win-9.3-51.5\bin`)

   b. `<root directory extracted from the Windows NSWL client package zip file>\etc` (for example, `C:\nswl_win-9.3-51.5\ etc`)

   c. `< root directory extracted from the Windows NSWL client package zip file >\samples` (for example, `C:\nswl_win-9.3-51.5\ samples`)

3. At a command prompt, run the following command from the `<root directory extracted from the Windows NSWL client package zip file>\bin path:`

   nswl **-install -f** <directorypath> \log.conf

   `<directorypath>:` Specifies the path to the configuration file (`log.conf`). By default, `log.conf` is in the `< root directory extracted from the Windows NSWL client package zip file >\samples` directory. But you can copy log.conf to your desired directory.

To uninstall the NSWL client on a Windows system

At a command prompt, run the following from the <root directory extracted from the Windows NSWL client package zip file>\bin path:

nswl -**remove**

Installing NSWL Client on an AIX Operating System

Copy the installation files from the NetScaler product CD or download them from ftp.netscaler.com. Log on to the AIX system as an administrator. Use the following procedure to install the NSWL executable and the other files on the system.

To install the NSWL client package on an AIX operating system

1. Copy the NSweblog.rpm file into a temporary directory:

   cp <path_to_cd>/Utilities/weblog/AIX/NSweblog.rpm /tmp

2. To install the NSWL executable, use the following command:

   rpm -i NSweblog.rpm

   This command extracts the files and installs them in the following directories.

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

To uninstall the NSWL client package on an AIX operating system

At a command prompt, type:

rpm -e NSweblog

To get more information about the NSweblog RPM file

At a command prompt, type:

rpm -qpi *.rpm

To view the installed Web server logging files

At a command prompt, type:

rpm -qpl *.rpm

NSWL Client Command Options

The following table describes the commands that you can use to configure the NSWL client.

Table 1. NSWL Command Options

NSWL command	Specifies
nswl -help	The available NSWL help options.
nswl -addns -f <path to configuration file>	The system that gathers the log transaction data. You are prompted to enter the IP address of the NetScaler appliance. Enter a valid user name and password.
nswl -verify -f <path to configuration file>	Check for syntax or semantic errors in the configuration file (for example, log.conf).
nswl -start -f <path to configuration file>	Start the NSWL client based on the settings in the configuration file (for example, log.conf). For Solaris and Linux: To start Web server logging as a background process, type the ampersand sign (&) at the end of the command.
nswl -stop (Solaris and Linux only)	Stop the NSWL client if it was started as a background process; otherwise, use CTRL+C to stop Web server logging.
nswl -install -f <path to configuration file> (Windows only)	Install the NSWL client as a service in Windows.
nswl -startservice (Windows only)	Start the NSWL client by using the settings in the configuration file (for example, log.conf) specified in the nswl install option. You can also start NSWL client from **Start** > **Control Panel** > **Services**.
nswl -stopservice (Windows only)	Stops the NSWL client.
nswl -remove	Remove the NSWL client service from the registry.

Run the following commands from the directory in which the NSWL executable is located:

- Windows: \ns\bin
- Solaris and Linux: \usr\local\netscaler\bin

The Web server logging configuration files are located in the following directory path:

- Windows: \ns\etc

- Solaris and Linux: \usr\local\netscaler\etc

The NSWL executable is started as .\nswl in Linux and Solaris.

Adding the IP Addresses of the NetScaler Appliance

In the NSWL client configuration file (log.conf), add the NetScaler IP address (NSIP) from which the NSWL client will start collecting logs.

To add the NSIP address of the NetScaler appliance

1. At the client system command prompt, type:

   nswl -addns -f < directorypath > \log.conf

   < directorypath >: Specifies the path to the configuration file (log.conf).

2. At the next prompt, enter the following information:

   - NSIP: Specify the IP address of the NetScaler appliance.
   - User name: Specify the user name of the NetScaler appliance.
   - Password: Specify the password.

Note: If you add multiple NetScaler IP addresses (NSIP), and later you do not want to log all of NetScaler system log details, you can delete the NSIPs manually by removing the NSIP statement at the end of the log.conf file. During a failover setup, you must add both primary and secondary NetScaler IP addresses to the log.conf by using the command. Before adding the IP address, make sure the user name and password exist on the NetScaler appliances.

Verifying the NSWL Configuration File

To make sure that logging works correctly, check the NSWL configuration file (log.conf) on the client system for syntax errors.

To verify the configuration in the NSWL configuration file

At the client system command prompt, type:

nswl -**verify** -**f** <directorypath>**log.conf**

< directorypath >: Specifies the path to the configuration file (log.conf).

Running the NSWL Client

To start Web server logging

At the client system command prompt, type:

nswl -**start** -**f** <directorypath>\log.conf

<directorypath>: Specifies the path to the configuration file (log.conf).

To stop Web server logging started as a background process on the Solaris or Linux operating systems

At the command prompt, type:

nswl -**stop**

To stop Web server logging started as a service on the Windows operating system

At the command prompt, type:

nswl -**stopservice**

Customizing Logging on the NSWL Client System

You can customize logging on the NSWL client system by making additional modifications to the NSWL client configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the client system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information based on the the host IP address, domain name, and host name of the Web servers.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

Creating Filters

You can use the default filter definition located in the configuration file (log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

Note: Consolidated logging, which logs transactions for which no filter is defined, uses the default filter if it is enabled. Consolidated logging of all servers can be done by defining only the default filter.

If the server hosts multiple Web sites and each Web site has its own domain name, and each domain is associated with a virtual server, you can configure Web server logging to create a separate log directory for each Web site. The following table displays the parameters for creating a filter.

Table 1. Parameters for Creating a Filter

Parameter	Specifies
filterName	Name of the filter (maximum 64 alphanumeric characters).
HOST name	Host name of the server for which the transactions are being logged.
IP ip	IP address of the server for which transactions are to be logged (for example, if the server has multiple domains that have one IP address).
IP ip 2...ip n:	Multiple IP addresses (for example, if the server domain has multiple IP addresses).
ip6 ip	IPv6 address of the server for which transactions are to be logged.
IP ip NETMASK mask	IP addresses and netmask combination to be used on a subnet.
ON | OFF	Enable or disable the filter to log transactions. If no argument is selected, the filter is enabled (ON).

To create a filter

To create a filter, enter the following command in the log.conf file:

- filter <filterName> <HOST name> | [**IP** <ip>] | [**IP** <ip 2...ip n>] | <IP ip NETMASK mask> [ON | OFF]
- filter <filterName> <HOST name> | [**IP6 ip**/<prefix length>] [ON | OFF]

To create a filter for a virtual server

To create a filter for a virtual server, enter the following command in the log.conf file:

filter <filterName> <VirtualServer IP address>

Example

In the following example, you specify an IP address of 192.168.100.0 and netmask of 255.255.255.0. The filter applies to IP addresses 192.168.100.1 through 192.168.100.254.

Filter F1 HOST www.netscaler.com ON
Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
Filter F4 IP 192.168.100.151
Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com IP 192.168.100.200 ON
Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
For creating filters for servers having IPv6 addresses.
Filter F9 2002::8/112 ON
Filter F10 HOST www.abcd.com IP6 2002::8 ON

Specifying Log Properties

Log properties are applied to all log entries associated with the filter. The log property definition begins with the keyword BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
 logFormat ...
 logFilenameFormat ...
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
```

Entries in the definition can include the following:

- **LogFormat** specifies the Web server logging feature that supports NCSA, W3C Extended, and custom log file formats.

  By default, the logformat property is w3c. To override, enter custom or NCSA in the configuration file, for example:

  ```
  LogFormat NCSA
  ```

  Note: For the NCSA and custom log formats, local time is used to time stamp transactions and for file rotation.

- **LogInterval** specifies the intervals at which new log files are created. Use one of the following values:

  - Hourly: A file is created every hour.
  - Daily: A file is created every day at midnight. Default value.
  - Weekly: A file is created every Sunday at midnight.
  - Monthly: A file is created on the first day of the month at midnight.
  - None: A file is created only once, when Web server logging starts.

  Example

  ```
  LogInterval Daily
  ```

- **LogFileSizeLimit** specifies the maximum size of the log file in MB. It can be used with any log interval (weekly, monthly, and so on.) A file is created when the maximum file size limit is reached or when the defined log interval time elapses.

  To override this behavior, specify the size as the loginterval property so that a file is created only when the log file size limit is reached.

  The default LogFileSizeLimit is 10 MB.

  Example

  ```
  LogFileSizeLimit 35
  ```

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:

  - Static: Specifies a constant string that contains the absolute path and file name.

    Dynamic: Specifies an expression containing the following format:

    - Server IP address (%A)
    - Date (%{format}t)
    - URL suffix (%x)
    - Host name (%v)

  Example

  ```
  LogFileNameFormat Ex%{%m%d%y}t.log
  ```

  This command creates the first file name as Exmmddyy.log, then every hour creates a file with file name: Exmmddyy.log.0, Exmmddyy.log.1,..., Exmmddyy.log.n.

  Example

  ```
  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%{%m%d%y}t
  ```

  Caution: The date format %t specified in the LogFilenameFormat command overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat.

- **LogExclude** prevents logging of transactions with the specified file extensions.

Example

```
LogExclude .html
```

This command creates a log file that excludes log transactions for *.html files.

- **LogTime** specifies log time as either GMT or LOCAL.

  The defaults are:

  - NCSA log file format: LOCAL
  - W3C log file format: GMT.

Understanding the NCSA and W3C Log Formats

The NetScaler supports the following standard log file formats:

- NCSA Common Log Format
- W3C Extended Log Format

NCSA Common Log Format

If the log file format is NCSA, the log file displays log information in the following format:

Client_IP_address -User_Name [Date:Time -TimeZone] "Method Object HTTP_version" HTTP_StatusCode Byte

To use the NCSA Common log format, enter NCSA in the LogFormat argument in the log.conf file.

The following table describes the NCSA Common log format.

Table 1. NCSA Common Log Format

Argument	Specifies
Client _IP_address	The IP address of the client computer.
User Name	The user name.
Date	The date of the transaction.
Time	The time when the transaction was completed.
Time Zone	The time zone (Greenwich Mean Time or local time).
Method	The request method (for example; GET, POST).
Object	The URL.
HTTP_version	The version of HTTP used by the client.
HTTP_StatusCode	The status code in the response.
Bytes Sent	The number of bytes sent from the server.

W3C Extended Log Format

An extended log file contains a sequence of lines containing ASCII characters terminated by either a Line Feed (LF) or the sequence Carriage Return Line Feed (CRLF.) Log file generators must follow the line termination convention for the platform on which they are run.

Log analyzers must accept either LF or CRLF form. Each line may contain either a directive or an entry. If you want to use the W3C Extended log format, enter W3C as the Log-Format argument in the log.conf file.

By default, the standard W3C log format is defined internally as the custom log format, shown as follows:

```
%{%Y-%m-%d%H:%M:%S}t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{user-agent}i %+{cookie} i%+{referer}i
```

For a description of the meaning of this each custom format, see Appendix A: Arguments for Defining a Custom Log Format. You can also change the order or remove some fields in this W3C log format. For example:

```
logFormat W3C %{%Y-%m-%d%H:%M:%S}t %m %U
```

W3C log entries are created with the following format:

```
#Version: 1.0
#Fields: date time cs-method cs-uri
#Date: 12-Jun-2001 12:34
2001-06-12 12:34:23 GET /sports/football.html
2001-06-12 12:34:30 GET /sports/football.html
```

Entries

Entries consist of a sequence of fields relating to a single HTTP transaction. Fields are separated by white space; Citrix recommends the use of tab characters. If a field in a particular entry is not used, a dash (-) marks the omitted field.

Directives

Directives record information about the logging process. Lines beginning with the pound sign (#) contain directives.

The following table describes the directives.

Table 2. Directive Descriptions

Directive	Description
Version: <integer>.<integer>	Displays the version of the extended log file format used. This document defines version 1.0.
Fields: [<specifier>...]	Identifies the fields recorded in the log.
Software: <string>	Identifies the software that generated the log.
Start-Date: <date> <time>	Displays the date and time at which the log was started.
End-Date: <date> <time>	Displays the date and time at which logging finished.
Date: <date> <time>	Displays the date and time when the entry was added.
Remark: <text>	Displays comments. Analysis tools ignore data recorded in this field.

Note: The Version and Fields directives are required. They precede all other entries in the log file.

Example

The following sample log file shows the log entries in W3C Extended log format:

```
#Version: 1.0
#Fields: time cs-method cs-uri
#Date: 12-Jan-1996 00:00:00
00:34:23 GET /sports/football.html
12:21:16 GET /sports/football.html
12:45:52 GET /sports/football.html
12:57:34 GET /sports/football.html
```

Fields

The Fields directive lists a sequence of field identifiers that specify the information recorded in each entry. Field identifiers may have one of the following forms:

- **identifier:** Relates to the transaction as a whole.
- **prefix-identifier:** Relates to information transfer between parties defined by the value *prefix*.
- **prefix (header):** Specifies the value of the HTTP header field header for transfer between parties defined by the value *prefix*. Fields specified in this manner always have the type <string>.

The following table describes defined prefixes.

Table 3. Prefix Descriptions

Prefix	Specifies
c	Client
s	Server
r	Remote
cs	Client to server
sc	Server to client
sr	Server to remote server (prefix used by proxies)
rs	Remote server to server (prefix used by proxies)
x	Application-specific identifier

Examples

The following examples are defined identifiers that use prefixes:

cs-method: The method in the request sent by the client to the server.

sc(Referer): The Referer field in the reply.

c-ip: The IP address of the client.

Identifiers

The following table describes the W3C Extended log format identifiers that do not require a prefix.

Table 4. W3C Extended Log Format Identifiers (No Prefix Required)

Identifier	Description
date	The date on which the transaction was done.
time	The time when the transaction is done.
time-taken	The time taken (in seconds) for the transaction to complete.
bytes	The number of bytes transferred.
cached	Records whether a cache hit has occurred. A zero indicates a cache miss.

The following table describes the W3C Extended log format identifiers that require a prefix.

Table 5. W3C Extended Log Format Identifiers (Requires a Prefix)

Identifier	Description
IP	The IP address and the port number.
dns	The DNS name.
status	The status code.
comment	The comment returned with status code.
method	The method.
url	The URL.
url-stem	The stem portion of the URL.
url-query	The query portion of the URL.

The W3C Extended Log file format allows you to choose log fields. These fields are shown in the following table.

Table 6. W3C Extended Log File Format (Allows Log Fields)

Field	Description
Date	The date on which the transaction is done.
Time	The time when the transaction is done.
Client IP	The IP address of the client.
User Name	The user name.
Service Name	Theservice name, which is always HTTP.
Server IP	The server IP address.
Server Port	The server port number
Method	The request method (for example; GET, POST).
Url Stem	The URL stem.
Url Query	The query portion of the URL.
Http Status	The status code in the response.
Bytes Sent	The number of bytes sent to the server (request size, including HTTP headers).
Bytes Received	The number of bytes received from the server (response size, including HTTP headers).
Time Taken	The time taken for transaction to complete, in seconds.
Protocol Version	The version number of HTTP being used by the client.
User Agent	The User-Agent field in the HTTP protocol.
Cookie	The Cookie field of the HTTP protocol.
Referer	The Referer field of the HTTP protocol.

Creating a Custom Log Format

You can customize the display format of the log file data manually or by using the NSWL library. By using the custom log format, you can derive most of the log formats that Apache currently supports.

Creating a Custom Log Format by Using the NSWL Library

Use one of the following NSWL libraries depending on whether the NSWL executable has been installed on a Windows or Solaris host computer:

- **Windows:** The nswl.lib library located in \ns\bin directory on the system manager host computer.
- **Solaris:** The libnswl.a library located in /usr/local/netscaler/bin.

To create the custom log format by using the NSWL Library

1. Add the following two C functions defined by the system in a C source file:

   ns_userDefFieldName() : This function returns the string that must be added as a custom field name in the log record.

   ns_userDefFieldVal() : This function implements the custom field value, then returns it as a string that must be added at the end of the log record.

2. Compile the file into an object file.
3. Link the object file with the NSWL library (and optionally, with third party libraries) to form a new NSWL executable.
4. Add a %d string at the end of the logFormat string in the configuration file (log.conf).

Example

```
##########
# A new file is created every midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log and create digital
#signature field for each record.
BEGIN CACHE_F
   logFormat   custom "%a - "%{user-agent}i" [%d/%B/%Y %T -%g] "%x" %s %b%{referrer}i "%{user-agent}i" "%{coo
   logInterval         Daily
```

```
    logFileSizeLimit        20
    logFilenameFormat       /datadisk5/netscaler/log/%v/NS%{%m%d%y}t.log
END CACHE_F
```

Creating a Custom Log Format Manually

To customize the format in which log file data should appear, specify a character string as the argument of the LogFormat log property definition. For more information, see Appendix A: Arguments for Defining a Custom Log Format. The following is an example where character strings are used to create a log format:

```
 LogFormat Custom ""%a - "%{user-agent}i" %[%d/%m/%Y]t %U %s %b %T"
```

- The string can contain the “c” type control characters \n and \t to represent new lines and tabs.
- Use the <Esc> key with literal quotes and backslashes.

The characteristics of the request are logged by placing % directives in the format string, which are replaced in the log file by the values.

If the %v (Host name) or %x (URL suffix) format specifier is present in a log file name format string, the following characters in the file name are replaced by an underscore symbol in the log configuration file name:

" * . / : < > ? \ |

Characters whose ASCII values lie in the range of 0-31 are replaced by the following:

%<ASCII value of character in hexadecimal>.

For example, the character with ASCII value 22 is replaced by %16.

Caution: If the %v format specifier is present in a log file name format string, a separate file is opened for each virtual host. To ensure continuous logging, the maximum number of files that a process can have open should be sufficiently large. See your operating system documentation for a procedure to change the number of files that can be opened.

Creating Apache Log Formats

You can derive from the custom logs most of the log formats that Apache currently supports. The custom log formats that match Apache log formats are:

NCSA/combined: LogFormat custom %h %l %u [%t] "%r" %s %B "%{referer}i" "%{user-agent}i"

NCSA/Common: LogFormat custom %h %l %u [%t] "%r" %s %B

Referer Log: LogFormat custom "%{referer}i" -> %U

Useragent: LogFormat custom %{user-agent}i

Similarly, you can derive the other server log formats from the custom formats.

Sample Configuration File

Following is a sample configuration file:

```
##########
# This is the NSWL configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##########
##########
# Default filter (default on)
# W3C Format logging, new file is created every hour or on reaching 10MB file size,
# and the file name is Exyymmdd.log
##########
Filter default
begin default
	logFormat		W3C
	logInterval		Hourly
	logFileSizeLimit	10
	logFilenameFormat	Ex%{%y%m%d}t.log
end default
##########
# netscaler caches example
# CACHE_F filter covers all the transaction with HOST name www.netscaler.com and the listed server ip's
##########
#Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95 192.168.100.52 192.168.100.53
##########
# netscaler origin server example
# Not interested in Origin server to Cache traffic transaction logging
##########
#Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66 192.168.100.67 192.168.100.225 1
100.227 192.168.100.228 OFF
##########
# netscaler image server example
# all the image server logging.
##########
#Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71 192.168.100.72 192.168.100.169
0.171 ON
##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log.
# Exclude objects that ends with .gif .jpg .jar.
##########
#begin ORIGIN_SERVERS
#	logFormat		NCSA
#	logInterval		Daily
#	logFileSizeLimit	40
#	logFilenameFormat	/datadisk5/ORGIN/log/%v/NS%{%m%d%y}t.log
#	logExclude		.gif .jpg .jar
```

```
#end ORIGIN_SERVERS

##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log with log record timestamp as
##########
#begin CACHE_F
#       logFormat             NCSA
#       logInterval           Daily
#       logFileSizeLimit       20
#       logFilenameFormat /datadisk5/netscaler/log/%v/NS%{%m%d%y}t.log
#       logtime               GMT
#end CACHE_F

##########
# W3C Format logging, new file on reaching 20MB and the log file path name is
# atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record timestamp as LOCAL.
##########
#begin IMAGE_SERVER
#       logFormat             W3C
#       logInterval           Size
#       logFileSizeLimit       20
#       logFilenameFormat /datadisk6/netscaler/log/%AEx%{%m%d%y}t
#       logtime               LOCAL
#end IMAGE_SERVER

##########
# Virtual Host by Name firm, can filter out the logging based on the host name by,
##########

#Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
#begin VHOST_F
#       logFormat             W3C
#       logInterval           Daily
#       logFileSizeLimit       10
logFilenameFormat /ns/prod/vhost/%v/Ex%{%m%d%y}t
#end VHOST_F

########## END FILTER CONFIGURATION ##########
```

Arguments for Defining a Custom Log Format

The following table describes the data that you can use as the Log Format argument string:

Table 1. Custom Log Format

Argument	Specifies
%a	The remote IPv4 address.
%A	The local IPv4 address.
%a6	The remote IPv6 address.
%A6	The local IPv6 address.
%B	The bytes sent, excluding the HTTP headers (response size).
%b	The bytes received, excluding the HTTP headers (request size).
%d	A user-defined field.
%g	The Greenwich Mean Time offset (for example, -0800 for Pacific Standard Time).
%h	The remote host.
%H	The request protocol.
%{Foobar}i	The contents of the Foobar: header line(s) in the request sent to the server. The system supports the User-Agent, Referer and cookie headers. The + after the % in this format informs the logging client to use the + as a word separator.
%j	The bytes received, including headers (request size)
%J	The bytes sent, including headers (response size)
%l	The remote log name (from identd, if supplied).
%m	The request method.
%M	The time taken to serve the request (i n microseconds)
%{Foobar}o	The contents of Foobar: header line(s) in the reply. We support the USER-AGENT, Referer, and cookie headers.

%p	The canonical port of the server serving the request.
%q	The query string (prefixed with a question mark (?) if a query string exists).
%r	The first line of the request.
%s	For requests that were redirected internally, this is the status of the original request.
%t	The time, in common log format (standard English time format).
%{format}t	The time, in the form given by format, must be in the strftime(3) format. For format descriptions, see Appendix B: Time Format Definition.
%T	The time taken to serve the request, in seconds.
%u	The remote user (from auth; may be bogus if return status (%s) is 401).
%U	The URL path requested.
%v	The canonical name of the server serving the request.
%V	This is the virtual server IPv4 address in the system, if load balancing, content switching, and/or cache redirection is used.
%V6	This is the virtual server IPv6 address in the system, if load balancing, content switching, and/or cache redirection is used.

For example, if you define the log format as %+{user-agent}i, and if the user agent value is Citrix NetScaler system Web Client, then the information is logged as Citrix NetScaler system +Web+Client. An alternative is to use double quotation marks. For example, “%{user-agent}i” logs it as “Citrix NetScaler system Web Client.” Do not use the <Esc> key on strings from %.. .r, %. . .i and, %. . .o. This complies with the requirements of the Common Log Format. Note that clients can insert control characters into the log. Therefore, you should take care when working with raw log files.

Time Format Definition

The following table lists the characters that you can enter as the format part of the %{format}t string described in the Custom Log Format table of Arguments for Defining a Custom Log Format. Values within brackets ([ ]) show the range of values that appear. For example, [1,31] in the %d description in the following table shows %d ranges from 1 to 31.

Table 1. Time Format Definition

Argument	Specifies
%%	The same as %.
%a	The abbreviated name of the week day for the locale.
%A	The full name of the week day for the locale.
%b	The abbreviated name of the month for the locale.
%B	The full name of the month for the locale.
%C	The century number (the year divided by 100 and truncated to an integer as a decimal number [1,99]); single digits are preceded by a 0.
%d	The day of month [1,31]; single digits are preceded by 0.
%e	The day of month [1,31]; single digits are preceded by a blank.
%h	The abbreviated name of the month for the locale.
%H	The hour (24-hour clock) [0,23]; single digits are preceded by a 0.
%I	The hour (12-hour clock) [1,12]; single digits are preceded by a 0.
%j	The number of the day in the year [1,366]; single digits are preceded by 0.
%k	The hour (24-hour clock) [0,23]; single digits are preceded by a blank.
%l	The hour (12-hour clock) [1,12]; single digits are preceded by a blank.
%m	The number of the month in the year [1,12]; single digits are preceded by a 0.

%M	The minute [00,59]; leading 0 is permitted but not required.
%n	Inserts a new line.
%p	The equivalent of either a.m. or p.m. for the locale.
%r	The appropriate time representation in 12-hour clock format with %p.
%S	The seconds [00,61]; the range of values is [00,61] rather than [00,59] to allow for the occasional leap second and for the double leap second.
%t	Inserts a tab.
%u	The day of the week as a decimal number [1,7]. 1 represents Sunday, 2 represents Tuesday and so on.
%U	The number of the week in the year as a decimal number [00,53], with Sunday as the first day of week 1.
%w	The day of the week as a decimal number [0,6]. 0 represents Sunday.
%W	Specifies the number of the week in the year as a decimal number [00,53]. Monday is the first day of week 1.
%y	The number of the year within the century [00,99]. For example, 5 would be the fifth year of that century.
%Y	The year, including the century (for example, 1993).

Note: If you specify a conversion that does not correspond to any of the ones described in the preceding table, or to any of the modified conversion specifications listed in the next paragraph, the behavior is undefined and returns 0.

The difference between %U and %W (and also between modified conversions %OU and %OW) is the day considered to be the first day of the week. Week number 1 is the first week in January (starting with a Sunday for %U, or a Monday for %W). Week number 0 contains the days before the first Sunday or Monday in January for %U and %W.

Advanced Configurations

You can configure network time protocol to synchronize a Citrix® NetScaler® appliance's local clock with the other servers on the network. If you enable path maximum transmission unit (PMTU) discovery, the NetScaler can use it to determine the maximum transmission unit of any Internet channel. For more efficient data transfer, you can configure TCP window scaling and selective acknowledgment. You can clear any basic or extended configuration on your NetScaler. You can view statistics associated with HTTP request and response sizes. For applying a specific HTTP and TCP settings to vservers and services, you can configure HTTP and TCP profiles.

Configuring Clock Synchronization

You can configure your NetScaler appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network.

You can configure clock synchronization on your appliance by adding NTP server entries to the ntp.conf file from either the configuration utility or the NetScaler command line, or by manually modifying the ntp.conf file, and then starting the NTP daemon (NTPD). The clock synchronization configuration does not change if the appliance is restarted, upgraded, or downgraded. However, the configuration does not get propagated to the secondary NetScaler in a high availability setup.

Note: If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site, http://www.ntp.org, under Public Time Servers List. Before configuring your NetScaler to use a public NTP server, be sure to read the Rules of Engagement page (link included on all Public Time Servers pages).

Setting Up Clock Synchronization by Using the CLI or the Configuration Utility

To configure clock synchronization from the configuration utility or from the CLI, you add NTP servers and then enable NTP synchronization.

To add an NTP server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an NTP server and verify the configuration:

- add ntp server (<serverIP> | <serverName>) [-**minpoll** <positive_integer>][-**maxpoll** <positive_integer>]
- show ntp server

Example

```
> add ntp server 1.2.3.5 -minpoll 6 -maxpoll 11
 Done
> sh ntp server

      NTP Server: xyz.net
      Minimum Poll Interval: 6 (64secs)
      Maximum Poll Interval: 9 (512secs)

      NTP Server: 1.2.3.5
      Minimum Poll Interval: 6 (64secs)
      Maximum Poll Interval: 11 (2048secs)
 Done
```

To modify or remove NTP servers by using the NetScaler command line

- To modify settings for an NTP server, type the set ntp server (<serverIP> | <serverName>) command and the parameters to be changed, with their new values.
- To remove an NTP server, type rm ntp server (<serverIP> | <serverName>)

Parameters for configuring an NTP server

serverIP

IP address of the NTP server.

serverName

Domain name of the NTP server.

minpoll

Minimum number of seconds after which the NTP server must poll the NTP messages, expressed as a power of 2. Minimum value: 4 (2^4=16 seconds). Maximum value: 17 (2^17=131072 seconds). Default: 6 (2^6=64 seconds).

maxpoll

Maximum number of seconds after which the NTP server must poll the NTP messages, expressed as a power of 2. Minimum value: 4 (2^4=16 seconds). Maximum value: 17 (2^17=131072 seconds). Default : 10 (2^10=1024 seconds).

To configure an NTP server by using the configuration utility

1. In the navigation pane, expand **System**, and then click **NTP Servers**.
2. In the details pane, do one of the following:
   - To add a new NTP server, click **Add**.
   - To modify settings for an existing NTP server, select the NTP server, and then click **Open**.
3. In the **Create NTP Server** or **Configure NTP Server** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring an NTP server” as shown:
   - NTP Server*—serverIP or serverName (Cannot be changed for an existing NTP server.)
   - Minimum Poll—minpoll
   - Maximum Poll—maxpoll

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the NTP server has been configured successfully.

Starting or Stopping the NTP Daemon

When you enable NTP synchronization, the NetScaler starts the NTP daemon and uses the NTP server entries in the ntp.conf file to synchronize its local time setting. If you do not want to synchronize your NetScaler time with the other servers in the network, you can disable NTP synchronization, which stops the NTP daemon (NTPD).

To enable or disable NTP synchronization by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable ntp sync
- disable ntp sync

To enable or disable NTP synchronization by using the configuration utility

1. In the navigation pane, expand **System**, and then click **NTP Servers**.
2. On the **NTP Servers** page, click **NTP Synchronization - ON** or **NTP Synchronization - OFF**.

Configuring Clock Synchronization Manually

You can configure clock synchronization manually by logging on to the NetScaler and editing the ntp.conf file.

To enable clock synchronization on your NetScaler by modifying the ntp.conf file

1. Log on to the NetScaler command line.
2. Switch to the shell prompt.
3. Copy the /etc/ntp.conf file to /nsconfig/ntp.conf, unless the /nsconfig directory already contains an ntp.conf file.
4. Check the /nsconfig/ntp.conf file for the following entries and, if they are present, remove them:

   restrict localhost

   restrict 127.0.0.2
5. Add the IP address for the desired NTP server to the /nsconfig/ntp.conf file, beneath the file's server and restrict entries.

   Note: For security reasons, there should be a corresponding restrict entry for each server entry.
6. If the /nsconfig directory does not contain a file named rc.netscaler, create the file.
7. Add the following entry to /nsconfig/rc.netscaler: /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &

   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   This process runs every time the NetScaler is restarted.
8. Reboot the NetScaler to enable clock synchronization.

   Note:

   If you want to start the time synchronization process without restarting the NetScaler, run the following command from the shell prompt:

   /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &

Viewing the System Date and Time

To change the system date and time, you must use the shell interface to the underlying FreeBSD OS. However, to view the system date and time, you can use the NetScaler command line or the configuration utility.

To view the system date and time by using the NetScaler command line

At the NetScaler command prompt, type:

show ns config

Example

```
> show ns config
        NetScaler IP: 10.102.29.170  (mask: 255.255.255.0)
        Number of MappedIP(s): 6
        Node: Standalone

        Global configuration settings:
                        HTTP port(s): (none)
                    Max connections: 0
          Max requests per connection: 0
                 Client IP insertion: DISABLED
                     Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                      Min Path MTU: 576
               Path MTU entry timeout: 10
                     FTP Port Range: 0
                      CR Port Range: 0
                          Timezone: GMT+05:30-IST-Asia/Colombo
                       System Time: Tue Feb 22 16:50:44 2011
                Last Config Changed Time: Tue Feb 22 16:48:02 2011
                 Last Config Saved Time: Tue Feb 22 16:48:19 2011
 Done
```

To view the system date and time by using the configuration utility

1. In the navigation pane, click **System**.
2. In the details pane, select the **System Information** tab.
3. Under **System Information**, view the system date and time.

Configuring TCP Window Scaling

The TCP window scaling option, which is defined in RFC 1323, increases the TCP receive window size beyond its maximum value of 65,535 bytes. This option is required for efficient transfer of data over long fat networks (LFNs).

A TCP window determines the amount of outstanding (unacknowledged by the recipient) data a sender can send on a particular connection before receiving any acknowledgment from the receiver. The main purpose of the window is flow control.

The window size field in the TCP header is 16 bits, which limits the ability of the sender to advertise a window size larger than 65535 (2^16 - 1). The TCP window scale extension expands the definition of the TCP window by applying a scale factor to the value in the 16 bit window size field of the TCP header. (Although RFC 1323 describes expanding the definition to up to 30 bits, NetScaler window scaling expands the definition of the TCP window to up to 24 bits.) The scale factor is carried in the new TCP window scale field. This field is sent only in a SYN packet (a segment with the SYN bit on)

To fit a larger window size value into the 16-bit field, the sender right shifts the value by the number of bit positions specified by the scale factor. The receiver left shifts the value by the same number of positions. Therefore, the actual window size is equivalent to:

(2^<scale factor>) * <received window size>.

Before configuring window scaling, make sure that:

- You do not set a high value for the scale factor, because this could have adverse effects on the NetScaler and the network.
- You have enabled selective acknowledgement (SACK).
- You do not configure window scaling unless you clearly know why you want to change the window size.
- Both hosts in the TCP connection send a window scale option during connection establishment. If only one side of a connection sets this option, windows scaling is not used for the connection.
- Each connection for same session is an independent Window Scaling session. For example, when a client's request and the server's response flow through the NetScaler appliance, it is possible to have window scaling between the client and the appliance without window scaling between the appliance and the server.

By default, window scaling is not enabled.

To configure window scaling by using the NetScaler command line

At the NetScaler command prompt, type the following command to configure window scaling and verify the configuration:

- set ns tcpParam [-**WS** (ENABLED | DISABLED)] [-**WSVal** <positive_integer>]
- show ns tcpParam

Example

```
> set ns tcpParam -WS ENABLED  -WSVal 6
 Done
> sh ns tcpParam
       TCP Parameters

       Window Scaling status     : ENABLED
       Window Scaling factor     : 6
       SACK status                 : ENABLED
        .
      .
      .
     TCP minimum RTO in millisec: 1000
       TCP Slow start increment: 2
 Done
```

Parameters for configuring window scaling

WSVal

Factor used to calculate new window size. Possible values: 0 to 8. Default: 4.

WS

Enables or disables window scaling.

To configure window scaling by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Configure TCP Parameters**.
3. In the **Configure TCP Parameters** dialog box, under **TCP**, select the **Windows Scaling** check box to enable window scaling.
4. In the **Factor** text box, type a windows scaling factor (for example, **6**). For possible values, see “Parameters for configuring window scaling.”
5. Click **OK**. A message appears in the status bar, stating that window scaling has been configured successfully.

Configuring Selective Acknowledgment

NetScaler appliances support Selective Acknowledgment (SACK), as defined in RFC 2018. Using SACK, the data receiver (either a NetScaler or a client) notifies the sender about all the segments that have been received successfully. As a result, the sender (either a NetScaler or a client) needs to retransmit only those segments that were lost during transmission. This improves the performance of data transmission. SACK is important in long fat networks (LFNs). By default, SACK is disabled.

To enable Selective Acknowledgment (SACK) by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable Selective Acknowledgment (SACK) and verify the configuration:

- set ns tcpParam [-**SACK** (ENABLED | DISABLED)]
- show ns tcpParam

Example

```
> set ns tcpParam -SACK ENABLED
 Done
> show ns tcpParam
        TCP Parameters

        Window Scaling status     : ENABLED
        Window Scaling factor     : 4
        SACK status             : ENABLED
        MaxBurst setting          : 6 MSS
        Initial cwnd setting      : 4 MSS
        TCP Receive Buffer        : 8190 bytes
        TCP Delayed-ACK Timer     : 200 millisec
        Down Service Reset status : DISABLED
        Nagle's Algorithm : DISABLED
        Limited Persist Probes : ENABLED
        Maximum out-of-order packets to queue: 64
 Done
```

To enable SACK by using the Configuration Utility

1. In the navigation pane, expand **System**, and click **Settings**.

2. In the details pane, under **Settings**, click **Change TCP Parameters**.

3. In the **Configure TCP Parameters** dialog box, under **TCP**, select the **Selective Acknowledgment** check box, and then click **OK.** A message appears in the status bar, stating that SACK has been configured successfully.

Clearing the Configuration

You have the following three options for clearing your NetScaler configuration.

Basic level. Clearing your configuration at the basic level clears all settings except the following:

- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions
- Feature and mode settings
- Default administrator password (nsroot)

Extended level. Clearing your configuration at the extended level clears all settings except the following:

- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions

Feature and mode settings revert to their default values.

Full level. Clearing your configuration at the full level returns all settings to their factory default values. However, the NSIP and default gateway are not changed, because changing them could cause the NetScaler to lose network connectivity.

To clear a configuration by using the NetScaler command line

At the NetScaler command prompt, type the following command:

clear ns config < (basic | advanced | full)>

Example

> clear ns config basic

```
Are you sure you want to clear the configuration(Y/N)? [N]:Y
 Done
```

Parameters for clearing a configuration

level

A level representing the extent to which to clear the configuration. Possible values: basic, extended, full.

To clear a configuration by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Diagnostics**.
2. In the details pane, under **Maintenance**, click **Clear Configuration**.
3. In the **Clear Configuration** dialog box, for **Configuration Level**, select an option (for example, **basic**).
4. Click **Run**. A message appears in the status bar, stating that the configuration has been refreshed successfully.

Viewing the HTTP Band Statistics

You can view HTTP band statistics to obtain useful information such as:

- Average request/response band size.
- The size range to which most requests/responses belong.
- Contribution of HTTP pages, in a certain size range, to the overall HTTP traffic.

To view HTTP request and response size statistics by using the NetScaler command line

At the NetScaler command prompt, type:

show protocol httpBand **-type** (REQUEST|RESPONSE)

Example

```
show protocol httpBand -type REQUEST
show protocol httpBand -type RESPONSE
```

To view HTTP request and response size statistics by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **HTTP data band statistics**.
3. In the **HTTP Data Band Statistics** dialog box, view the HTTP request and HTTP response size statistics on the **Request** and **Response** tabs, respectively.

You can also modify the band range for HTTP request or response size statistics.

To modify the band range by using the NetScaler command line

At the NetScaler command prompt, type:

set protocol httpBand **reqBandSize** <value> **respBandSize** <value>

Example

set protocol httpBand reqBandSize 300 respBandSize 2048

Parameters for modifying the band range for HTTP request or response size statistics

reqBandSize

Band size for HTTP request band statistics, in bytes. Minimum value: 50. Maximum value: 2147483647. Default: 100.

respBandSize

Band size for HTTP response band statistics, in bytes. Minimum value: 50. Maximum value: 2147483647. Default: 1024.

To modify the band range by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **HTTP data band Statistics**. Do one or both of the following:

   To modify the band range of HTTP request statistics, click the **Request** tab,

   click the **Request** tab, and then click **Configure**. In **Change HTTP Request Band Size** dialog box, enter a value (for example, **300**) in **Request Data Band Size** text field, and then click **OK**.

   - To modify the band range of HTTP response statistics, click the **Response** tab, and then click **Configure**. In **Change HTTP Response Band Size** dialog box, enter a value (for example, **2048**) in **Response Data Band Size** text field, and then click **OK**.
3. Click **Close**.

Configuring HTTP Profiles

An HTTP profile is a collection of HTTP parameter settings that can be applied to virtual servers and services. An HTTP profile can be reused on multiple virtual servers or services.

You can use built-in HTTP profiles or configure custom profiles. The following table describes the built-in HTTP profiles.

Table 1. Built-in HTTP Profiles

Built-in profile	Description
nshttp_default_strict_validation	This profile is useful for deployments where strict validation of HTTP requests and responses is required.
nshttp_default_profile	This profile represents the default global HTTP settings on the NetScaler appliance.

To add an HTTP profile by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns httpProfile name **-maxReusePool** <value> **-dropInvalReqs** (ENABLED | DISABLED) **-markHttp09Inval** (ENABLED | DISABLED) **-markConnReqInval** (ENABLED | DISABLED) **-cmpOnPush** (ENABLED | DISABLED) **-conMultiplex** (ENABLED | DISABLED)
- sh ns httpProfile

Example

```
add ns httpProfile http_profile1 -maxReusePool 30 -dropInvalReqs ENABLED -markHttp09Inval ENABLED
-markConnReqInval ENABLED -cmpOnPush ENABLED -conMultiplex DISABLED
```

Parameters for adding an HTTP profile

name

The name for an HTTP profile. An HTTP profile name can be from 1 to 127 characters and must begin with a letter, a number, or the underscore symbol (_). Other characters allowed after the first character in a name are the hyphen (-), period (.), pound sign (#), space ( ), at sign (@), and equals sign (=).

maxReusePool

A maximum limit on the number of connections, from the NetScaler to a particular server, in the reuse pool. This setting is helpful for optimal memory utilization and for reducing the idle connections to the server just after the peak time.

conMultiplex

When this option is enabled, the NetScaler appliance reuse the server connections for multiple client connections. Possible values: ENABLED, DISABLED. Default: ENABLED.

dropInvalReqs

Drop the invalid request or responses, either based on the header or body. Possible values: ENABLED, DISABLED. Default: DISABLED.

markHttp09Inval

Mark the 0.9 requests as invalid or not. Possible values: ENABLED, DISABLED. Default: DISABLED.

markConnReqInval

Mark the CONNECT requests invalid or not. Possible values: ENABLED, DISABLED. Default: DISABLED.

cmpOnPush

Use compression on PUSH for the HTTP VIP. Possible values: ENABLED, DISABLED. Default: DISABLED.

To add an HTTP profile by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Profiles**.
2. In the details pane, on the **HTTP Profiles** tab, click **Add**.
3. In the **Create HTTP Profile** dialog box, specify values for the following parameters, which correspond to parameters described in " Parameters for adding an HTTP profile" as shown:
   - **Name***—name
   - **Max Connection in reusepool**—maxReusePool
   - **Connection Multiplexing**—conMultiplex
   - **Drop invalid HTTP requests**—dropInvalReqs
   - **Mark HTTP/0.9 requests as invalid**—markHttp09Inval
   - **Mark CONNECT requests as invalid**—markConnReqInval
   - **Compression on PUSH packet**—cmpOnPush

   * A required parameter.
4. Click **Create**. A message appears in the status bar, stating that the HTTP profile has been configured successfully.

Configuring TCP Profiles

A Transmission Control Protocol (TCP) profile is a collection of TCP parameter settings that can be applied to virtual servers and services. A TCP profile can be reused on multiple virtual servers or services. You can use built-in TCP profiles or configure custom profiles. The following table describes the built-in TCP profiles.

Table 1. Built-in TCP Profiles

Built-in profile	Description
nstcp_default_tcp_lfp	This profile is useful for long fat pipe networks (WAN) on the client side. Long fat pipe networks have long delay, high bandwidth lines with minimal packet drops.
nstcp_default_tcp_lnp	This profile is useful for long narrow pipe networks (WAN) on the client side. Long narrow pipe networks have considerable packet loss once in a while.
nstcp_default_tcp_lan	This profile is useful for back-end server connections, where these servers reside on the same LAN as the NetScaler appliance.
nstcp_default_tcp_lfp_thin_stream	This profile is similar to the nstcp_default_tcp_lfp profile; however, the settings are tuned for small size packet flows.
nstcp_default_tcp_lnp_thin_stream	This profile is similar to the nstcp_default_tcp_lnp profile; however, the settings are tuned for small size packet flows.
nstcp_default_tcp_lan_thin_stream	This profile is similar to the nstcp_default_tcp_lan profile; however, the settings are tuned to small size packet flows.
nstcp_default_tcp_interactive_stream	This profile is similar to the nstcp_default_tcp_lan profile; however, it has a reduced delayed ACK timer and ACK on PUSH packet settings.
nstcp_internal_apps	This profile is useful for internal applications on the NetScaler appliance (for example, GSLB sitesyncing). This contains tuned window scaling and SACK options for the desired applications. This profile should not be bound to applications other than internal applications.

nstcp_defualt_profile	This profile represents the default global TCP settings on the NetScaler appliance.

To add a TCP profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a TCP profile and verify the configuration:

- add ns tcpProfile name **-WS** (ENABLED | DISABLED) **-SACK** (ENABLED | DISABLED) **-WSVal** <value> **-nagle** (ENABLED | DISABLED) **-ackOnPush** (ENABLED | DISABLED) **-maxBurst** value **-initialCwnd** <value> **-delayedAck** <value> **-oooQSize** <value> **-maxPktPerMss** <value> **-pktPerRetx** <value> **-minRTO** <value> **-slowStartIncr** <value>
- sh ns tcpProfile

Example

```
add ns tcpProfile tcp_profile1 -WS ENABLED -SACK ENABLED -WSVal 4 -nagle DISABLED
 -ackOnPush ENABLED -maxBurst 10 -initialCwnd 6 -delayedAck 200 -oooQSize 100
 -maxPktPerMss 0 -pktPerRetx 3 -minRTO 200 -slowStartIncr 3
```

Parameters for creating a TCP profile

name

The name for a TCP profile. A TCP profile name can be from 1 to 127 characters and must begin with a letter, a number, or the underscore symbol (_). Other characters allowed after the first character in a name are the hyphen (-), period (.), pound sign (#), space ( ), at sign (@), and equals sign (=).

WS

Enable or disable window scaling. Possible values: ENABLED, DISABLED. Default: DISABLED.

WSVal

The factor used to calculate the new window size. Possible values: 0 to 8. Default: 4.

maxBurst

The maximum number of TCP segments allowed in a burst. Minimum value: 2. Maximum value: 10. Default: 6.

initialCwnd

The initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server. Minimum value: 2. Maximum value: 6. Default: 4.

delayedAck

The time-out for TCP delayed ACK, in milliseconds. Minimum value: 10. Maximum value: 200. Default: 300.

oooQSize

The maximum size of out-of-order packets queue. Minimum value: 0 0 means infinite). Maximum value: 512. Default: 64.

maxPktPerMss

The maximum number of TCP packets allowed per maximum segment size (MSS). Minimum value: 0. Maximum value: 1460. Default: 0 Means that no maximum is set.)

pktPerRetx

The maximum limit on the number of packets that should be retransmitted on receiving a partial ACK. Minimum value: 1. Maximum value: 100. Default: 1.

minRTO

The minimum round trip to origin (RTO) time, in milliseconds. Minimum value: 10. Maximum value: 64,000. Default: 1,000.

slowStartIncr

The multiplier that determines the rate at which slow start increases the size of the TCP transmission window after each acknowledgement of successful transmission. Minimum value: 1. Maximum value: 100. Default: 2.

SACK

Enable or disable selective acknowledgement (SACK). Possible values: ENABLED, DISABLED. Default: DISABLED.

nagle

Enable or disable the Nagle algorithm on TCP connections. Possible values: ENABLED, DISABLED. Default: DISABLED.

ackOnPush

Send immediate positive acknowledgement (ACK) on receipt of TCP packets when doing Web 2.0 PUSH. Possible values: ENABLED, DISABLED. Default: ENABLED.

To add a TCP profile by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Profiles**.
2. In the details pane, on the **TCP Profiles** tab, click **Add**.
3. In the **Create TCP Profiles** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a TCP profile" as shown:
   - **Name***—name
   - **Window Scaling**—WS
   - **Factor**—WSVal
   - **Maximum Burst Limit**—maxBurst
   - **Initial Congestion Window Size**—initialCwnd
   - **TCP Delayed ACK Time-out (msec)**—delayedAck
   - **Maximum ooo Packet Queue Size**-oooQSize
   - **Maximum Packets Per MSS**—maxPktPerMss
   - **Maximum Packets per Retransmission**—pktPerRetx
   - **Minimum RTO (in millisec)**—minRTO
   - **Slow Start Increment**—slowStartIncr
   - **Selective Acknowledgement**—SACK
   - **Use Nagle's Algorithm**—nagle
   - **Immediate ACK on Receiving Packet with PUSH**—ackOnPush

   * A required parameter.
4. Click **Create**. A message appears in the status bar, stating that the TCP profile has been configured successfully.

Specifying a TCP Buffer Size

You can set the TCP buffer size, both globally and for individual virtual servers and services, through TCP profiles. The value that you set is the minimum value that is advertised by the NetScaler appliance, and this buffer size is reserved when a client initiates a connection that is associated with an endpoint-application function, such as compression or SSL. The managed application can request a larger buffer, but if it requests a smaller buffer, the request is not honored, and the specified buffer size is used. If the TCP buffer size is set both at the global level and at the entity level (virtual server or service level), the buffer specified at the entity level takes precedence. If the buffer size that you specify for a service is not the same as the buffer size that you specify for the virtual server to which the service is bound, the NetScaler appliance uses the buffer size specified for the virtual server for the client-side connection and the buffer size specified for the service for the server-side connection. However, for optimum results, make sure that the values specified for a virtual server and the services bound to it have the same value. The buffer size that you specify is used only when the connection is associated with endpoint-application functions, such as SSL and compression.

You set the TCP buffer size in a custom, entity-level TCP profile by setting the `bufferSize` parameter for the profile. To apply the buffer size setting specified in a custom, entity-level profile, you bind the profile to the virtual server or service. You set the global TCP buffer size by setting the `bufferSize` parameter in the global TCP profile `nstcp_default_profile`. You do not bind `nstcp_default_profile` to an entity. The settings in `nstcp_default_profile` are automatically applied globally.

Note: A high TCP buffer value could limit the number of connections that can be made to the NetScaler appliance. Additionally, the global TCP parameter `recvBuffSize`, which was set by the use of the `set ns tcpParam` command, has been deprecated. You can now specify the buffer size only through TCP profiles.

To set the TCP buffer size in an entity-level TCP profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the TCP buffer size in a TCP profile and verify the configuration:

```
set ns tcpProfile <name> -bufferSize <positive_integer>

show ns tcpProfile <name>
```

Example

```
> set ns tcpProfile profile1 -bufferSize 12000
 Done
```

```
> show ns tcpProfile profile1
      Name    : profile1
      Window Scaling status     : DISABLED
      Window Scaling factor     : 4
      .
      .
      .
      TCP Buffer Size     : 12000 bytes
      Reference count: 0

 Done
>
```

To set the TCP buffer size in the global TCP profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the TCP buffer size in the global TCP profile and verify the configuration:

- **set ns tcpProfile nstcp_default_profile –bufferSize** <positive_integer>
- **show ns tcpProfile nstcp_default_profile**

Example

```
> set ns tcpProfile nstcp_default_profile -bufferSize 12000
 Done
> show ns tcpProfile nstcp_default_profile
      Name    : nstcp_default_profile
      Window Scaling status     : DISABLED
      Window Scaling factor     : 4
      .
      .
      .
      TCP Buffer Size     : 12000 bytes
      Reference count: 200

 Done
>
```

Parameters for setting the TCP buffer size in a TCP profile

name

Name of the TCP profile. Maximum length: 127 characters.

bufferSize

TCP buffer size in bytes. Maximum value: 4194304. Minimum value: 8190. Default: 8190.

To set the TCP buffer size in a TCP profile by using the NetScaler configuration utility

1. In the navigation pane, expand **System**, and then click **Profiles**.
2. In the details pane, click the **TCP Profiles** tab, and then do one of the following:
   - To create a custom, entity-level TCP profile, click **Add** and, in the **Create TCP Profile** dialog box, type a name for the new profile.
   - To set the TCP buffer size for an existing TCP profile, click the name of the TCP profile, and then click **Open**. If you want to set the TCP buffer size in the global TCP profile, click **nstcp_default_profile**.
3. In the **Create TCP Profile** or **Configure TCP Profile** dialog box, in the **TCP Buffer Size (Bytes)** box, type the number of bytes to specify as the minimum TCP buffer size.
4. Click **Create** or **OK**.

Optimizing the TCP Maximum Segment Size for a Virtual Server Configuration

You can specify the Maximum Segment Size (MSS) that the Citrix® NetScaler® appliance advertises to a client when the client initiates a connection to a virtual server on the appliance. You can configure the MSS for the virtual servers configured on the appliance in two ways:

- You can set the MSS for each virtual server to a value of your choice in a TCP profile.

  You can set the `learnVsvrMSS` global TCP parameter to `ENABLED` to enable MSS learning for all the virtual servers configured on the appliance.

If you know the optimal MSS value for a given virtual server, you can specify the MSS in a TCP profile and bind the profile to the virtual server. When a client initiates a connection with the virtual server, the NetScaler appliance advertises the specified MSS value to the client. However, if the appliance is also configured to learn the optimum MSS value from bound services (as described in the following section), the learned MSS value takes precedence, and the value specified in the TCP profile is used only until the appliance learns the optimum MSS value. The appliance uses the learned MSS value until the appliance is restarted. If the appliance is restarted, the appliance defaults to the MSS value specified in the virtual server's TCP profile until it learns the MSS value again.

Specifying the MSS Value in a TCP Profile

If you know the optimal MSS value for a given virtual server, you can specify the MSS in a TCP profile and bind the profile to the virtual server. When a client initiates a connection with the virtual server, the NetScaler appliance advertises the specified MSS value to the client.

To specify the MSS value in a TCP profile by using the NetScaler command-line

At the NetScaler command prompt, type the following commands to specify the MSS value in a TCP profile and verify the configuration:

- **add ns tcpProfile** <name> **-mss** <positive_integer>
- **show ns tcpProfile**

```
> add ns tcpProfile tcp_prof1 -mss 1000
 Done
> show ns tcpProfile tcp_prof1
       Name    : tcp_prof1
       Window Scaling status     : DISABLED
       Window Scaling factor     : 4
       SACK status             : DISABLED
       MSS                   : 1000
       MaxBurst setting        : 6 MSS
       Initial cwnd setting      : 4 MSS
       .
       .
       .

 Done
>
```

Parameters for specifying the MSS value in a TCP profile

name

The name of the TCP profile.

mss

The maximum number of octets to allow in a TCP data segment.

To specify the MSS value in a TCP profile by using the NetScaler configuration utility

1. In the navigation pane, expand **System**, and then click **Profiles**.
2. In the details pane, do one of the following:
   - To create a TCP profile, click **Add**.
   - To specify the MSS in an existing TCP profile, click the name of the profile, and then click **Open**.
3. In the **Create TCP Profile** or **Configure TCP Profile** dialog box, specify values for the following parameters, which correspond to the parameters described in "Parameters for specifying the MSS value in a TCP profile" as shown:
   - **Name***—name (cannot be changed for an existing TCP profile)
   - **MSS***—mss

   * A required parameter
4. Click **Create** or **OK**.

Configuring the NetScaler to Learn the MSS Value from Bound Services

If you set the global TCP parameter `learnVsvrMSS` to `ENABLED`, the NetScaler appliance learns the most frequently used MSS value for each configured virtual server. When a client connects to a virtual server, the appliance advertises to the client the MSS value that is optimum for that virtual server. The optimum value is the MSS of the service or subset of bound services that are most frequently selected during load balancing. Consequently, each virtual server configuration uses its own MSS value. This enhancement enables the appliance to optimize the consumption of system resources.

The default value of the `learnVsvrMSS` parameter is `DISABLED`. When enabled, MSS learning is applicable only to virtual servers of type TCP, HTTP, and FTP.

To configure the NetScaler to learn the MSS for a virtual server by using the NetScaler command-line

At the NetScaler command prompt, type the following commands to configure the NetScaler to learn the MSS for a virtual server and verify the configuration:

- **`set ns tcpParam -learnVsvrMSS`** `(ENABLED|DISABLED)`
- **`show ns tcpParam`**

Example

```
> set ns tcpParam -learnVsvrMSS ENABLED
 Done
> show ns tcpParam
        TCP Parameters

        Window Scaling status     : DISABLED
        Window Scaling factor     : 4
        SACK status               : DISABLED
        Learn MSS for VServer     : ENABLED
        .
        .
        .
 Done
>
```

Parameters for configuring the NetScaler to learn the MSS for a virtual server

learnVsvrMSS

Enable or disable MSS learning for virtual servers. Possible values: `ENABLED`, `DISABLED`. Default: `DISABLED`.

To configure the NetScaler to learn the MSS for a virtual server by using the NetScaler configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, click **Change TCP parameters**.
3. In the **Configure TCP Parameters** dialog box, select the **Learn MSS** check box.

Web Interface

The Web Interface on Citrix® NetScaler® appliances is based on Java Server Pages (JSP) technology and provides access to Citrix® XenApp™ and Citrix® XenDesktop® applications. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in.

The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance. The Web Interface sites provide user access to the XenApp and XenDesktop resources, which include applications, content, and desktops.

Note: This feature is supported only on NetScaler nCore builds.

The Web Interface installation includes installing the Web Interface tar file and JRE tar file on the NetScaler appliance. To configure the Web Interface, you create a Web Interface site and bind one or more XenApp or XenDesktop farms to it.

How Web Interface Works

The following figure illustrates a basic Web interface session.

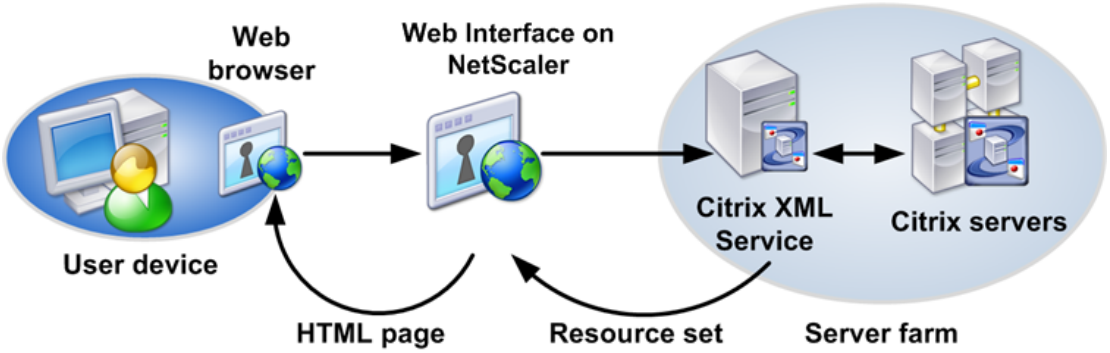


Figure 1. *A Basic Web Interface Session*

Following is a typical set of interactions among a user device, a NetScaler running the Web interface, and a server farm.

1. A user authenticates to the Web interface through a Web browser or by using the XenApp plug-in.
2. The Web interface reads the user's credentials and forwards the information to the Citrix XML Service running on servers in the server farm.
3. The Citrix XML Service on the designated server retrieves from the servers a list of resources that the user can access. These resources constitute the user's resource set and are retrieved from the Independent Management Architecture (IMA) system.
4. The Citrix XML Service then returns the user's resource set to the Web interface running on the NetScaler.
5. The user clicks an icon that represents a resource on the HTML page.
6. The Web interface queries the Citrix XML Service for the least busy server.
7. The Citrix XML Service returns the address of this server to the Web interface.
8. The Web interface sends the connection information to the Web browser.
9. The Web browser initiates a session with the server.

Prerequisites

The following prerequisites are required before you begin installing and configuring the Web interface.

- XenApp or XenDesktop farms are set up and running in your environment. For more information about XenApp, see the XenApp documentation at http://edocs.citrix.com/. For more information about XenDesktop, see the XenDesktop farms documentation at http://edocs.citrix.com/.

- Conceptual knowledge of the Web interface. For more information about Web interface running on a server, see the Web interface documentation at http://edocs.citrix.com/.

Installing the Web Interface

To install the Web interface, you need to install the following files:

- **Web interface tar file**. A setup file for installing the Web interface on the NetScaler. This tar file also includes Apache Tomcat Web server version 6.0.26. The file name has the following format: nswi-<version number>.tgz (for example, nswi-1.1.tgz).
- **JRE tar file**. This is Diablo Latte JRE version 1.6.0-7 for 64-bit FreeBSD 6.x/amd64 platform. To download the JRE tarball, see the FreeBSD Foundation Web site at http://www.freebsdfoundation.org/downloads/java.shtml.

Copy the tar files to a local workstation or to the /var directory of the NetScaler appliance.

These files install all the Web interface components and JRE on the NetScaler hard drive and configure automatic startup of the Tomcat Web server with Web interface at the NetScaler appliance startup time. Both tar files are internally expanded in the /var/wi directory on the hard drive.

To install the Web interface and JRE tar files by using the NetScaler command line

At the NetScaler command prompt, type:

install wi package **-wi** <URL> **-jre** <URL>

Example

install wi package -wi sftp://username:password@10.102.29.12/var/nswi-1.1.tgz -jre ftp://username:passw
install wi package -wi ftp://username:password@10.102.29.15/var/nswi-1.1.tgz -jre file:///var/diablo-jrefr

Parameters for installing the Web interface and JRE tar files

Web Interface tar file path

Complete path to the Web interface tar file.

JRE tar file path

Complete path to the JRE tar file.

To install the Web interface and JRE tar files by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Web Interface**.
2. In the details pane, under **Getting Started**, click **Install Web Interface**.
3. In the **Install Web Interface** dialog box, in the **Web Interface tar file path** text box, type the complete path to the Web interface tar file. You can also use the **browse** button to locate the file on your local system or the NetScaler hard drive.
4. In the **JRE tar file path** text box, type the complete path to the JRE tar file. You can also use the **browse** button to locate the file on your local system or the NetScaler hard drive.
5. Click **Install**.

Configuring the Web Interface

To configure the Web interface, you create a Web interface site and bind one or more XenApp or XenDesktop farms to it. You then configure the Web interface to work behind an HTTP or an HTTPS virtual server or Citrix Access Gateway™.

The following access methods are available for setting up Web interface sites:

Direct Mode. You create an HTTP or an HTTPS virtual server on the NetScaler appliance and bind the Web interface service, running on port 8080 of the NetScaler appliance, to the virtual server. Clients on the LAN use the virtual server IP address to access the Web interface. When using this access method, the URL format for the Web interface site is as follows:

```
<HTTP or HTTPS>://<HTTP or HTTPS vserver IP address>:<vserver port number>/<Web Interface site path>
```

Gateway Direct Mode. You associate the Web interface site with Access Gateway. Remote clients use the Access Gateway URL to access the Web interface site. With this access method, the URL format for the Web interface site is as follows:

```
HTTPS://<Access Gateway URL>/<Web Interface site path>
```

Parameters for configuring Web interface sites

Site Path

Path to the Web interface site. This parameter is required. Type a site path or select one of the following:

- /Citrix/XenApp/
- /Citrix/DesktopWeb/
- /Citrix/PNAgent/

Site Type

Type of site. Possible values: XenApp/XenDesktop Web Site (configures the Web interface site for access by a Web browser); XenApp/XenDesktop Services Site (configures the Web interface site for access by the XenApp plug-in). Default: XenApp/XenDesktop Web Site.

Published Resource Type

Method for accessing the published XenApp and XenDesktop resources. Possible values: Online (allows applications to be launched on the XenApp and XenDesktop servers); Offline (allows streaming of applications to the client); DualMode (allows both modes). Default: Online.

Direct Mode

The Web interface is accessed through an HTTP or an HTTPS loopback load balancing virtual server.

Virtual Server

Lists existing loopback load balancing virtual servers and an option to create a new one.

Protocol

The type of services to which the virtual server distributes requests. Possible values: HTTP, HTTPS. Default: HTTP.

IP Address

IP address of the virtual server. Can be an IPv4 or IPv6 address.

Port

Port on which the virtual server listens for client connections. Possible values: from 0 through 65535.

Gateway Direct Mode

The Web interface is accessed through a configured Access Gateway.

Authentication Point

Authentication point to be used for the site. Possible values: Web interface, AccessGateway. Default: AccessGateway.

Access Gateway URL

URL of the Access Gateway.

Add DNS Entry

Specifies whether to add DNS address record to resolve the specified Access Gateway URL. Possible values: ON, OFF. Default: ON.

Trust SSL Certificate

Specifies whether the Web interface site trusts certificates signed by a non-trusted CA. Possible values: ON, OFF. Default: ON.

STA Server URL

URL of the Secure Ticket Authority (STA) server.

STA Server URL (2)

URL of the second STA server.

Session Reliability

Specifies whether to use session reliability through the Access Gateway. Possible values: ON, OFF. Default: OFF.

Use Two STA Servers

Specifies whether the Web interface requests tickets from two separate gateway Secure Ticket Authorities when a resource is accessed. Possible values: ON, OFF. Default: OFF.

Kiosk Mode

Specifies whether user settings should be persistent or last only for the lifetime of the session. When Kiosk mode is enabled, user settings do not persist from one session to another. Possible values: ON, OFF. Default value: OFF.

Name

Name of a XenApp or XenDesktop farm. Any name can be used as a logical representation of a XenApp or XenDesktop farm. The name must not exceed 127 characters.

XML Service Addresses

Comma-separated IP addresses or host names of either XenApp or XenDesktop servers providing XML services.

XML Service Port

Port number to use for contacting the XML service. Default: 80.

Transport

Transport protocol to use for the XML service. Possible values: HTTP, HTTPS. Default: HTTP.

Load balance

Specifies whether to use all the XML servers (load balance mode) or only one (failover mode). Possible values: ON (load balance mode), OFF (failover mode). Default: ON.

Configuring a Web Interface Site for LAN Users Using HTTP

In this scenario, user and the Web interface setup are on the same enterprise LAN. The enterprise has both a XenApp and a XenDesktop farm. Users access the Web interface by using an HTTP vserver. The Web interface exposes its own login page for authentication. The vserver IP address is used to access the Web interface.

The following figure illustrates the Web interface running on the NetScaler appliance NS1. A Web interface site WINS1 is created and a XenApp farm XA1 and a XenDesktop farm XD1 are bound to it. An HTTP vserver HTTP_WI is also created. Client C1 uses the IP address of the HTTP_WI vserver to access the WINS1 site.

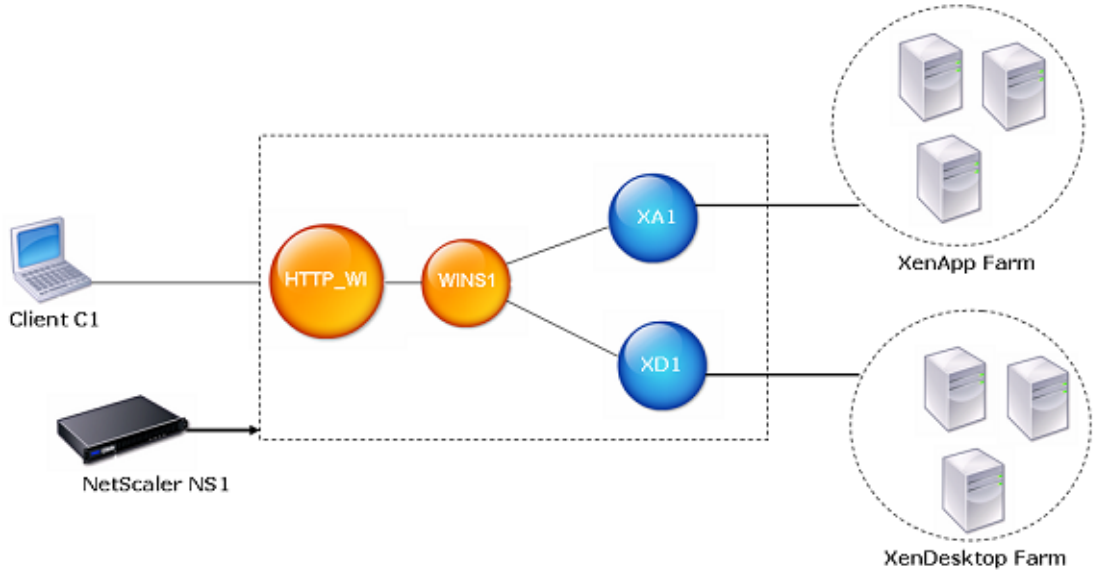


Figure 1. A Web Interface Site Configured for LAN Users Using HTTP

To configure a Web interface site for LAN users using HTTP by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Web Interface**.
2. In the details pane, click **Web Interface Wizard**.
3. On the wizard **Introduction** page, click **Next**.
4. On the wizard **Configure Web Interface Site** page, specify the values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Site Path*** (You cannot change the name of an existing Web interface site.)
   - **Site Type**
   - **Published Resource Type**
   - **Kiosk Mode**

   * A required parameter.
5. Select **Direct Mode** and specify values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Virtual Server**
   - **Protocol** (select **HTTP**)
   - **IP Address**
   - **Port**

   Note:

   When you create the HTTP vserver by using the configuration utility, the configuration utility automatically creates a service, which logically represents the Web interface service running on the NetScaler appliance, and binds the service to the HTTP virtual server.

   For more information about services and virtual servers, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.
6. Click **Next**.
7. On the wizard's **Configure XenApp/XenDesktop Farm** page, do one of the following:

   - To add a XenApp or XenDesktop farm, click **Add**.
   - To modify an existing XenApp or XenDesktop farm, select the farm, and then click **Open**.

8. In the **Create XenApp/XenDesktop Farm** or **Configure XenApp/XenDesktop Farm** dialog box, specify values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Name*** (You cannot change the name of an existing XenApp or XenDesktop farm.)
   - **XML Service Addresses***
   - **XML Service Port**
   - **Transport**
   - **Load Balance**

   * A required parameter.

9. Click **Next**, and then click **Finish**.

10. Verify that the Web interface site you configured is correct by selecting the site and viewing the **Details** section at the bottom of the pane. To view the Web interface site, in the navigation pane, expand **System**, expand **Web Interface**, and then click **Sites**.

To configure a Web interface site for LAN users using HTTP by using the command line

1. Add a Web interface site. At the NetScaler command prompt, type:

   add wi site <sitePath> **-siteType** (XenAppWeb | XenAppServices)**-publishedResourceType** (Online | Offline | DualMode) **-kioskMode** (ON | OFF)

   Example

   add wi site WINS1 -siteType XenAppWeb -publishedResourceType Online -kioskMode ON

2. Bind XenApp or XenDesktop farms to the Web interface site. At the NetScaler command prompt, type:

   bind wi site <sitePath> <farmName> <xmlServerAddresses> **-xmlPort** <value> **-transport** (HTTP | HTTPS) **-loadBalance** (ON | OFF)

   Example

   bind wi site WINS1 XA1 10.102.46.6 -xmlPort 80 -transport HTTP -LoadBalance OFF
   bind wi site WINS1 XD1 10.102.46.50 -xmlPort 80 -transport HTTP -LoadBalance OFF

3. Create a service that is a logical representation of the Web interface service running on the NetScaler appliance. At the NetScaler command prompt, type: add service <name> <IP address> <serviceType> <port> **Example**

   add service WI_Loopback_Service 127.0.0.1 HTTP 8080

   For more information, see the “Load Balancing” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

4. Add an HTTP vserver. At the NetScaler command prompt, type:

   add lb vserver <virtualServerName> <protocol> <IPAddress> <port>

   Example

   add lb vserver HTTP_WI HTTP 10.102.29.5 80

   For more information, see the “Load Balancing” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

5. Bind the Web interface service to the HTTP vserver. At the NetScaler command prompt, type:

   bind lb vserver <virtualServerName> <serviceName>

   Example

   bind lb vserver HTTP_WI WI_Loopback_Service

   For more information, see the “Load Balancing” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Configuring a Web Interface Site for LAN Users Using HTTPS

In this scenario, user accounts and the Web interface setup are on the same enterprise LAN. Users access the Web interface by using an SSL-based (HTTPS) vserver. The Web interface exposes its own login page for authentication. SSL offloading is done by this vserver on the NetScaler. The vserver IP address is used to access the Web interface instead of the NetScaler IP address (NSIP).

The following figure illustrates the Web interface running on the NetScaler appliance NS1. A Web interface site WINS1 is created and a XenApp farm XA1 and a XenDesktop farm XD1 are bound to it. An HTTPS vserver HTTPS_WI is also created. Client C1 uses the IP address of the HTTPS_WI vserver to access the WINS1 site.

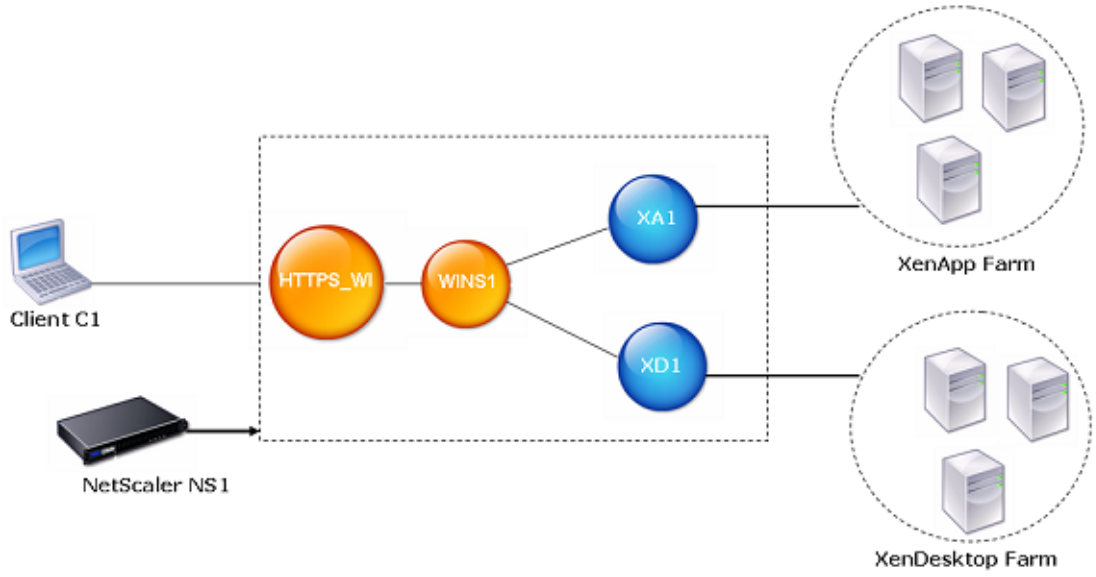


Figure 1. A Web Interface Site Configured for LAN Users Using HTTPS

To configure a Web interface site for LAN users using HTTPS by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Web Interface**.
2. In the details pane, click **Web Interface Wizard**.
3. On the wizard **Introduction** page, click **Next**.
4. On the wizard **Configure Web Interface Site** page, specify the values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Site Path*** (You cannot change the name of an existing Web interface site.)
   - **Site Type**
   - **Published Resource Type**
   - **Kiosk Mode**

   * A required parameter.
5. Select **Direct Mode** and specify values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Virtual Server**
   - **Protocol** (select **HTTPS**)
   - **IP Address**
   - **Port**

   Note:

   When you create the HTTPS vserver by using the configuration utility, the configuration utility automatically creates a service, which logically represents the Web interface service running on the NetScaler appliance, and binds the service to the HTTPS virtual server.

   For more information about services and virtual servers, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.
6. Click **Next**.
7. On the wizard's **Specify a server Certificate** page, you create or specify an existing SSL certificate-key pair. The SSL certificate-key pair is automatically bound to the HTTPS vserver.

   For more information, see “Binding an SSL Certificate Key Pair to the Virtual Server” in the “Secure Sockets Layer (SSL) Acceleration” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

8. On the wizard's **Configure XenApp/XenDesktop Farm** page, do one of the following:
   - To add a XenApp or XenDesktop farm, click **Add**.
   - To modify an existing XenApp or XenDesktop farm, select the farm, and then click **Open**.
9. In the **Create XenApp/XenDesktop Farm** or **Configure XenApp/XenDesktop Farm** dialog box, specify values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:
   - **Name*** (You cannot change the name of an existing XenApp or XenDesktop farm.)
   - **XML Service Addresses***
   - **XML Service Port**
   - **Transport**
   - **Load Balance**

   * A required parameter.
10. Click **Next**, and then click **Finish**.
11. Verify that the Web interface site you configured is correct by selecting the site and viewing the **Details** section at the bottom of the pane. To view the Web interface site, in the navigation pane, expand **System**, expand **Web Interface**, and then click **Sites**.

To configure a Web interface site for LAN users using HTTPS by using the command line

1. Add a Web interface site. At the NetScaler command prompt, type:

   add wi site <sitePath> **-siteType** (XenAppWeb | XenAppServices)**-publishedResourceType** (Online | Offline | DualMode) **-kioskMode** (ON | OFF)

   Example

   add wi site WINS1 -siteType XenAppWeb -publishedResourceType Online -kioskMode ON

2. Bind XenApp or XenDesktop farms to the Web interface site. At the NetScaler command prompt, type:

   bind wi site <sitePath> <farmName> <xmlServerAddresses> **-xmlPort** <value> **-transport** (HTTP | HTTPS) **-loadBalance** (ON | OFF)

   Example

   bind wi site WINS1 XA1 10.102.46.6 -xmlPort 80 -transport HTTP -LoadBalance OFF
   bind wi site WINS1 XD1 10.102.46.50 -xmlPort 80 -transport HTTP -LoadBalance OFF

3. Create a service that is a logical representation of the Web interface service running on the NetScaler appliance. At the NetScaler command prompt, type: add service <name> <IPAddress> <serviceType> <port> **Example**

   add service WI_Loopback_Service 127.0.0.1 HTTP 8080

   For more information, see the “Load Balancing” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

4. Add an HTTPS vserver. At the NetScaler command prompt, type:

   add lb vserver <virtualServerName> <protocol> <IPAddress> <port>

   Example

   add lb vserver HTTPS_WI SSL 10.102.29.3 443

   For more information, see “Adding an SSL-Based Virtual Server” in the “Secure Sockets Layer (SSL) Acceleration” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

5. Bind the Web interface service to the HTTPS vserver. At the NetScaler command prompt, type:

   bind lb vserver <virtualServerName> <serviceName>

   Example

   bind lb vserver HTTPS_WI WI_Loopback_Service

For more information, see “Binding Services to the Virtual Server” in the “Secure Sockets Layer (SSL) Acceleration” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

6. Create an SSL certificate key pair. At the NetScaler command prompt, type:

   add ssl certkey <certificate-KeyPairName> **-cert** <certificateFileName> **-key** <privateKeyFileName>

   Example

   add ssl certkey SSL-Certkey-1 -cert /nsconfig/ssl/test1.cer -key /nsconfig/ssl/test1

   For more information, see “Adding a Certificate Key Pair” in the “Secure Sockets Layer (SSL) Acceleration” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

7. Bind the SSL certificate key pair to the HTTPS vserver. At the NetScaler command prompt, type:

   bind ssl vserver <vserverName> **-certkeyName** <certificate- KeyPairName>

   Example

   bind ssl vserver HTTPS_WI -certkeyName SSL-Certkey-1

   For more information, see “Binding an SSL Certificate Key Pair to the Virtual Server” in the “Secure Sockets Layer (SSL) Acceleration” chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

8. Add a rewrite action. At the NetScaler command prompt, type:

   add rewrite action <name> <type> <target> [<stringBuilderExpr>] [(**-pattern** <expression>]

   Example

   add rewrite action Replace_HTTP_to_HTTPS INSERT_AFTER "HTTP.RES.HEADER(\"Location\").Value(0).Pre

   For more information, see “Configuring a Rewrite Action” in the “Rewrite” chapter of the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

9. Create a rewrite policy and bind the rewrite action to it. At the NetScaler command prompt, type:

   add rewrite policy <name> <expression> <rewriteAction>

   Example

   add rewrite policy rewrite_location "HTTP.RES.STATUS == 302 && HTTP.RES.HEADER(\"Location\").Value(

   For more information, see “Configuring a Rewrite Policy” in the “Rewrite” chapter of the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

10. Bind the rewrite policy to the HTTPS vserver. At the NetScaler command prompt, type:

bind lb vserver <VserverName> **-policyname** <rewritePolicyName> **-priority** <value> **-type** response

Example

bind lb vserver HTTPS_WI -policyname rewrite_location -priority 10 -type response

For more information, see “Binding a Rewrite Policy” in the “Rewrite” chapter of the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Configuring a Web Interface Site for Remote Users Using AGEE

In this scenario, user accounts and the Web interface setup are on different networks. Users access a Web interface site by using Access Gateway Enterprise Edition (AGEE) URL. SmartAccess is automatically enabled.

The following figure illustrates the Web interface running on the NetScaler appliance NS1. A Web interface site WINS1 is created and a XenApp farm XA1 and a XenDesktop XD1 are bound to it. An AGEE VPN vserver AGEE_WI is also configured. The client uses the AGEE URL of the AGEE_WI to access the WINS1 site.

For more information about configuring AGEE, see the AGEE documentation at http://edocs.citrix.com/.

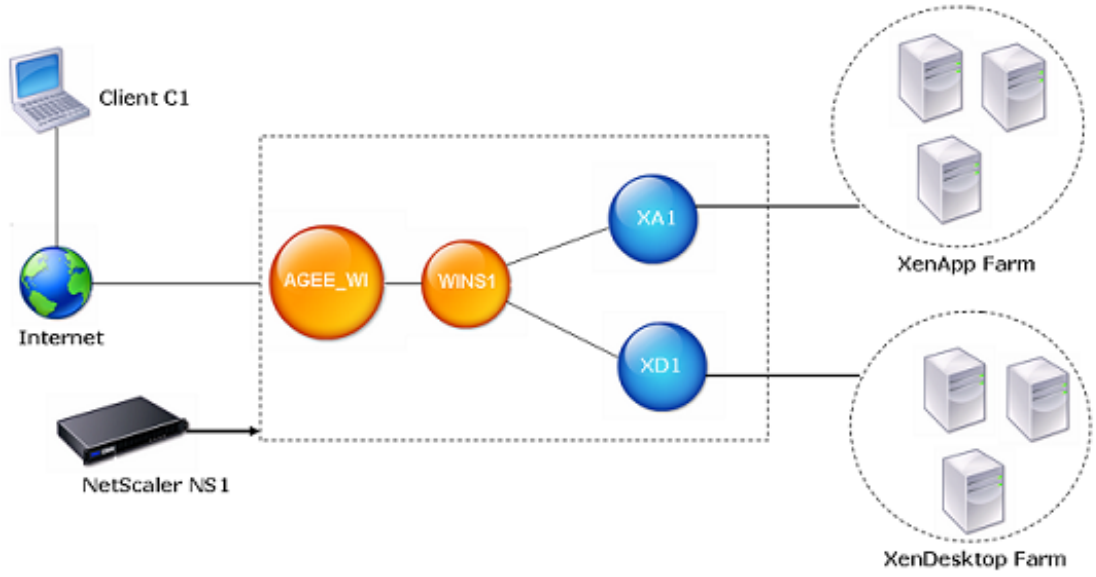


Figure 1. A Web Interface Site Configured for Remote Users Using AGEE

To configure a Web interface site for remote users using AGEE by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Web Interface**.
2. In the details pane, click **Web Interface Wizard**.
3. On the wizard **Introduction** page, click **Next**.
4. On the wizard **Configure Web Interface Site** page, specify the values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Site Path*** (You cannot change the name of an existing Web interface site.)
   - **Site Type**
   - **Published Resource Type**
   - **Kiosk Mode**

   * A required parameter.
5. Select **Gateway Direct Mode** and specify values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

   - **Authentication Point**
   - **Access Gateway URL**
   - **Add DNS Entry**
   - **Trust SSL Certificate**
   - **STA Server URL**
   - **STA Server URL (2)**
   - **Session Reliability**
   - **Use two STA Servers**
6. Click **Next**.
7. On the wizard's **Configure XenApp/XenDesktop Farm** page, do one of the following:

   - To add a XenApp or XenDesktop farm, click **Add**.
   - To modify an existing XenApp or XenDesktop farm, select the farm, and then click **Open**.
8. In the **Create XenApp/XenDesktop Farm** or **Configure XenApp/XenDesktop Farm** dialog box, specify values for the following parameters, which correspond to parameters described in Parameters for configuring Web interface sites as shown:

- **Name*** (You cannot change the name of an existing XenApp or XenDesktop farm.)
- **XML Service Addresses***
- **XML Service Port**
- **Transport**
- **Load Balance**

* A required parameter.

9. Click **Next**, and then click **Finish**.
10. Verify that the Web interface site you configured is correct by selecting the site and viewing the **Details** section at the bottom of the pane. To view the Web interface site, in the navigation pane, expand **System**, expand **Web Interface**, and then click **Sites**.

To configure a Web interface site for remote users using AGEE by using the command line

1. Add a Web interface site. At the NetScaler command prompt, type:

   add wi site <sitePath> <agURL> <staURL> **-sessionReliability** (ON | OFF) -useTwoTickets (ON | OFF) **-s**
   XenAppWeb | XenAppServices) **-publishedResourceType** (Online | Offline | DualMode) **-kioskMode** (O

   Example

   add wi site WINS1 https://ag.mycompany.com http://ag.staserver.com -sessionReliability OFF -authenti

2. Bind XenApp or XenDesktop farms to the Web interface site. At the NetScaler command prompt, type:

   bind wi site <sitePath> <farmName> <xmlServerAddresses> **-xmlPort** <value> **-transport** (HTTP | HTTPS)

   Example

   bind wi site WINS1 XA1 10.102.46.6 -xmlPort 80 -transport HTTP -LoadBalance OFF
   bind wi site WINS1 XD1 10.102.46.50 -xmlPort 80 -transport HTTP -LoadBalance OFF

Enhanced Application Visibility Using AppFlow

The Citrix® NetScaler® appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits the information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

AppFlow use actions and policies to send records for a selected flow to specific set of collectors. An AppFlow action specifies which set of collectors will receive the AppFlow records. Policies, which are based on Advanced expressions can be configured to select flows for which flow records will be sent to the collectors specified by the associated AppFlow action.

To limit the types of flows, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

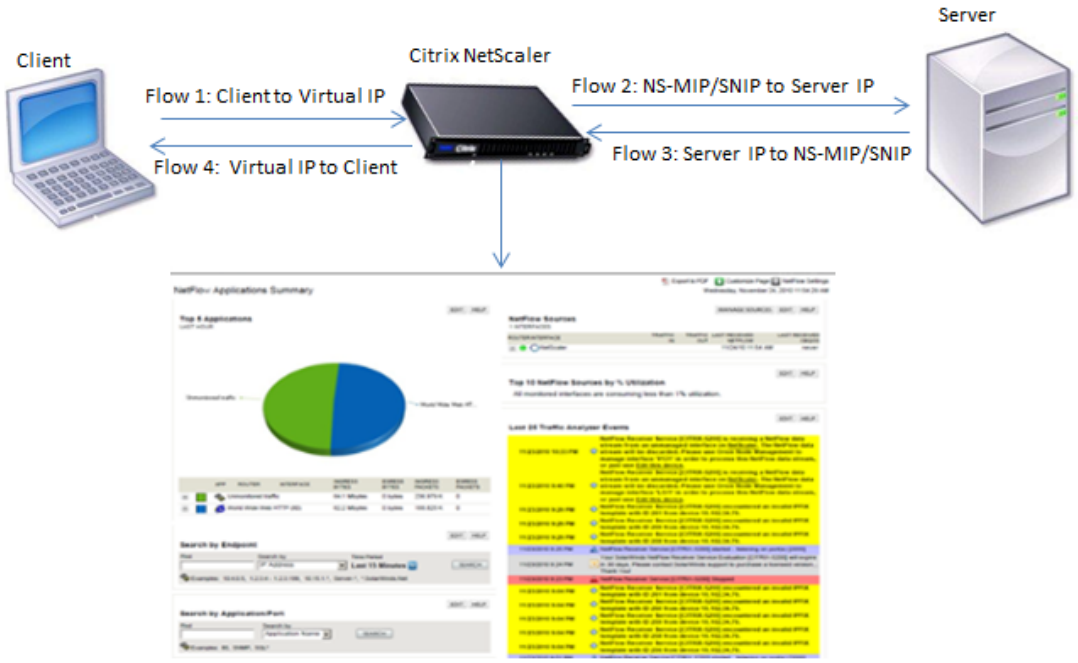
Note: This feature is supported only on NetScaler nCore builds.

How AppFlow Works

In the most common deployment scenario, inbound traffic flows to a Virtual IP address (VIP) on the NetScaler appliance and is load balanced to a server. Outbound traffic flows from the server to a mapped or subnet IP address on the NetScaler and from the VIP to the client. A flow is a unidirectional collection of IP packets identified by the following five tuples: sourceIP, sourcePort, destIP, destPort, and protocol.

The following figure describes how the AppFlow feature works.

Figure 1. NetScaler Flow Sequence



As shown in the figure, the network flow identifiers for each leg of a transaction depend on the direction of the traffic.

The different flows that form a flow record are:

Flow1: <Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

Flow2: <NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

Flow3: <Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

Flow4: <VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

To help the collector link all four flows in a transaction, AppFlow adds a custom transactionID element to each flow. For application-level content switching, such as HTTP, it is possible for a single client TCP connection to be load balanced to different backend TCP connections for each request. AppFlow provides a set of records for each transaction.

Flow Records

AppFlow records contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. AppFlow records also contain application-level information (such as HTTP URLs, HTTP request methods and response status codes, server response time, and latency). IPFIX flow records are based on templates that need to be sent before sending flow records.

Templates

AppFlow defines a set of templates, one for each type of flow. Each template contains a set of standard Information Elements (IEs) and Enterprise-specific Information Elements (EIEs). IPFIX templates define the order and sizes of the Information Elements (IE) in the flow record. The templates are sent to the collectors at regular intervals, as described in RFC 5101.

A template can include the following EIEs:

transactionID

An unsigned 32-bit number identifying an application-level transaction. For HTTP, this corresponds to a request and response pair. All flow records that correspond to this request and response pair have the same transaction ID. In the most common case, there are four uniflow records that correspond to this transaction. If the NetScaler generates the response by itself (served from the integrated cache or by a security policy), there may be only two flow records for this transaction.

connectionID

An unsigned 32-bit number identifying a layer-4 connection (TCP or UDP). The NetScaler flows are usually bidirectional, with two separate flow records for each direction of the flow. This information element can be used to link the two flows.

For the NetScaler, connectionID is an identifier for the connection data structure to track the progress of a connection. In an HTTP transaction, for instance, a given connectionID may have multiple transactionID elements corresponding to multiple requests that were made on that connection.

tcpRTT

The round trip time, in milliseconds, as measured on the TCP connection. This can be used as a metric to determine the client or server latency on the network.

httpRequestMethod

An 8-bit number indicating the HTTP method used in the transaction. An options template with the number-to-method mapping is sent along with the template.

httpRequestSize

An unsigned 32-bit number indicating the request payload size.

httpRequestURL

The HTTP URL requested by the client.

httpUserAgent

The source of incoming requests to the Web server.

httpResponseStatus

An unsigned 32-bit number indicating the response status code.

httpResponseSize

An unsigned 32-bit number indicating the response size.

httpResponseTimeToFirstByte

An unsigned 32-bit number indicating the time taken to receive the first byte of the response.

httpResponseTimeToLastByte

An unsigned 32-bit number indicating the time taken to receive the last byte of the response.

flowFlags

An unsigned 64-bit flag used to indicate different flow conditions.

Configuring the AppFlow Feature

You configure AppFlow in the same manner as most other policy-based features. First, you enable the AppFlow feature. Then you specify the collectors to which the flow records are sent. After that, you define actions, which are sets of configred collectors. Then you configure one or more policies and associate a action to each policy. The policy tells the NetScaler appliance to select requests the flow records of which are sent to the associated action. Finally, you bind each policy either globally or to a specific vservers to put it into effect.

You can further set AppFlow parameters to specify the template refresh interval and to enable the exporting of httpURL, httpCookie, and httpReferer information. On each collector, you must specify the NetScaler IP address as the address of the exporter.

Note: For information about configuring the NetScaler as an exporter on the collector, see the documentation for the specific collector.

The configuration utility provides tools that help users define the policies and actions that determine exactly how the NetScaler appliance export records for a particular flow to a set of collectors(action.) The NetScaler command line provides a corresponding set of CLI-based commands for experienced users who prefer a command line.

Enabling or Disabling the AppFlow Feature

To be able to use the AppFlow feature, you must first enable it.

To enable or disable the AppFlow feature by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- `enable ns feature appflow`
- `disable ns feature appflow`

To enable the AppFlow feature by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Configure advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **AppFlow** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Specifying a Collector

A collector receives flow records generated by the NetScaler appliance. To be able to send flow records, you must specify at least one collector. You can specify up to four. However, you cannot export the same data to multiple collectors. You can remove unused collectors.

To specify a collector by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a collector and verify the configuration:

- add appflowCollector <name> **-IPAddress** <ipaddress> **-port** <port_number>
- show appflowCollector <name>

Example

```
> add appflowCollector coll1 -IPAddress 10.102.29.251 -port 8000
Done

> show appflowCollector coll1

1)Collector name: coll1
      Collector IPv4 address: 10.102.29.251
      Collector UDP port: 8000
```

To remove a collector by using the NetScaler command line

At the NetScaler command prompt, type:

rm appflowCollector <name>

Parameters for specifying a collector

name

Name of the collector to which to export data. Maximum characters: 255.

ipaddress

The IPv4 address of the collector.

port

The UDP port on which the collector is listening. Default port: 4739.

To specify a collector by using the configuration utility

1. In the navigation pane, expand **AppFlow**, and then click **Collectors**.
2. In the details pane, click **Add**.
3. In the **Create AppFlow Collector** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for specifying a collector" as shown:
   - **Name***—name (cannot be changed for an existing collector)
   - **IP Address***—ipaddress (cannot be changed for an existing collector)
   - **Port**—port (cannot be changed for an existing collector)

   *A required parameter
4. To remove a collector from the list, select the collector, and then click **Remove**.
5. Click **Create**, and then click **Close**.

Setting the AppFlow Parameters

You can set AppFlow parameters to customize the exporting of data to the collectors.

To set the AppFlow Parameters by using the NetScaler Command Line

At the NetScaler command prompt, type the following commands to set the AppFlow parameters and verify the settings:

- set appflowParam [**-templateRefresh** <secs>] [-**appnameRefresh** <secs>] [-**flowRecordInterval** <secs>] [**-udpPmtu** <positive_integer>] [**-httpUrl** (**ENABLED** | **DISABLED**)] [**-httpCookie** (**ENABLED** | **DISABLED**)] [**-httpReferer** (**ENABLED** | **DISABLED**)] [**-httpMethod** (**ENABLED** | **DISABLED**)] [**-httpHost** (**ENABLED** | **DISABLED**)] [**-httpUserAgent** (**ENABLED** | **DISABLED**)] [**-clientTrafficOnly** (**YES** | **NO**)]
- show appflowParam

Example

```
> set appflowParam -templateRefresh 240 -udpPmtu 128 -httpUrl enabled
Done

> show appflowparam
 AppFlow parameters
 IPFIX template refresh interval: 600 seconds
 IPFIX UDP Path MTU: 1472 bytes
 HTTP URL logging: DISABLED
 HTTP cookie logging: DISABLED
 HTTP referer logging: DISABLED
 HTTP method logging: ENABLED
 HTTP host logging: ENABLED
 HTTP user-agent logging: ENABLED
 Log only client-side traffic: NO
   Done
```

To return AppFlow parameters to their default values by using the NetScaler command line

Type the unset appflowParam command and the names of the parameters to be returned to the default values.

AppFlow Parameters

templateRefresh

The refresh interval, in seconds, at which to export the template data. Because data transport is in the UDP protocol, the templates must be resent at regular intervals. Minimum value: 60. Maximum value: 3600. Default: 600.

appnameRefresh

Interval at which Appnames are sent to the configured collectors, in seconds. Minimum value: 60. Maximum value: 3600. Default: 600.

flowRecordInterval

Interval at which flow records are sent to the configured collectors, in seconds. Minimum value: 60. Maximum value: 3600. Default: 600.

udpPmtu

The maximum length of the UDP datagram. Default: 1472.

httpUrl

The http URL received by the NetScaler appliance from the client. Possible values: ENABLED, DISABLED. Default: DISABLED.

httpCookie

Include the cookie that was in the HTTP request received by the NetScaler appliance from the client. Possible values: ENABLED, DISABLED. Default: DISABLED.

httpReferer

Include the Web page that was last visited by the client. Possible values: ENABLED, DISABLED. Default: DISABLED.

httpMethod

Include the method that was specified in the HTTP request received by the NetScaler appliance from the client. Possible values: ENABLED, DISABLED. Default: DISABLED.

httpHost

Include the host identified in the HTTP request received by the NetScaler appliance from the client. Possible values: ENABLED, DISABLED. Default: DISABLED.

httpUserAgent

Include the client application through which the HTTP request was received by the NetScaler appliance. Possible values: ENABLED, DISABLED. Default: DISABLED.

clientTrafficOnly

Generate AppFlow records only for the traffic from the client. Possible values: YES, NO. Default: NO.

To set the AppFlow parameters by using the configuration utility

1. In the navigation pane, click **AppFlow**.
2. On the **AppFlow** landing page, under **Settings**, click **Change AppFlow Settings**.
3. In the **Configure AppFlow Settings** dialog box, specify values for the following parameters, which correspond to parameters described in "AppFlow Parameters" as shown:
   - **Template Refresh Interval**—templateRefresh
   - **AppName Refresh Interval**—appnameRefresh
   - **Flow Record Export Interval**—flowRecordInterval
   - **HTTP URL**—httpUrl
   - **HTTP Cookie**—httpCookie
   - **HTTP Referer**—httpReferer
   - **HTTP Method**—httpMethod
   - **HTTP Host**—httpHost
   - **HTTP User-Agent**—httpUserAgent
   - **Template Refresh Interval**—templateRefresh
   - **UDP Maximum Transmission Unit**—udpPmtu
4. Click **OK**, and then click **Close**.

Reporting Tool

Use the Citrix® NetScaler® Reporting tool to view NetScaler performance statistics data as reports. Statistics data are collected by the nscollect utility and are stored in a database. When you want to view certain performance data over a period of time, the Reporting tool pulls out specified data from the database and displays them in charts.

Reports are a collection of charts. The Reporting tool provides built-in reports as well as the option to create custom reports. In a report, you can modify the charts and add new charts. You can also modify the operation of the data collection utility, nscollect, and stop or start its operation.

Using the Reporting Tool

The Reporting tool is a Web-based interface accessed from the Citrix® NetScaler® appliance. Use the Reporting tool to display the performance statistics data as reports containing graphs. In addition to using the built-in reports, you can create custom reports, which you can modify at any time. Reports can have between one and four charts. You can create up to 256 custom reports.

To invoke the Reporting tool

1. Use the Web browser of your choice to connect to the IP address of the NetScaler (for example, http://10.102.29.170/). The Web Logon screen appears.
2. In the **User Name** text box, type the user name assigned to the NetScaler.
3. In the **Password** text box, type the password.
4. In the **Start in** drop-down box, select **Reporting**.
5. Click **Login**.

The following screen shots show the report toolbar and the chart toolbar, which are frequently referenced in this documentation.

Figure 1. *Report Toolbar*

Figure 2. *Chart Toolbar*

Working with Reports

You can plot and monitor statistics for the various functional groups configured on the NetScaler over a specified time interval. Reports enable you to troubleshoot or analyze the behavior of your appliance. There are two types of reports: built-in reports and custom reports. Report content for built-in or custom reports can be viewed in a graphical format or a tabular format. The graphical view consists of line, area, and bar charts that can display up to 32 sets of data (counters). The tabular view displays the data in columns and rows. This view is useful for debugging error counters.

The default report that is displayed in the Reporting tool is CPU vs. Memory Usage and HTTP Requests Rate. You can change the default report view by displaying the report you want as your default view, and then clicking **Default Report**.

Reports can be generated for the last hour, last day, last week, last month, last year, or you can customize the duration.

You can do the following with reports:

- Toggle between a tabular view of data and a graphical view of data.
- Change the graphical display type, such as bar chart or line chart.
- Customize charts in a report.
- Export the chart as an Excel comma-separated value (CSV) file.
- View the charts in detail by zooming in, zooming out, or using a drag-and-drop operation (scrolling).
- Set a report as the default report for viewing whenever you log on.
- Add or remove counters.
- Print reports.
- Refresh reports to view the latest performance data.

Using Built-in Reports

The Reporting tool provides built-in reports for frequently viewed data. Built-in reports are available for the following seven functional groups: System, Network, SSL, Compression, Integrated Cache, Access Gateway, and Application Firewall. By default, the built-in reports are displayed for the last day. However, you can view the reports for the last hour, last week, last month, or last year.

Note: You cannot save changes to built-in reports, but you can save a modified built-in report as a custom report.

To display a built-in report

1. In the left pane of the Reporting tool, under **Built-in Reports**, expand a group (for example, **SSL**).
2. Click a report (for example, **SSL** > **All Backend Ciphers**).

Creating and Deleting Reports

You can create your own custom reports and save them with user-defined names for reuse. You can plot different counters for different groups based on your requirements. You can create up to 256 custom reports.

You can either create a new report or save a built-in report as a custom report. By default, a newly created custom report contains one chart named **System Overview**, which displays the **CPU Usage** counter plotted for the last day. You can customize the interval and set the data source and time zone from the report toolbar. Within a report, you can use the chart toolbars to add, modify, or delete charts, as described in Working with Charts.

By default, newly created custom reports contain one chart named **System Overview** that displays a **CPU Usage** counter plotted for the last day.

To create a custom report

1. In the Reporting tool, on the report toolbar, click **Create**, or if you want to create a new custom report based on an existing report, open the existing report, and then click **Save As**.
2. In **Report Name** box, type a name for the custom report.
3. Do one of the following:
   - To add the report to an existing folder, in **Create in** or **Save in**, click the down arrow to choose an existing folder, and then click **OK**.
   - To create a new folder to store the report, click the **Click to add folder** icon, in **Folder Name**, type the name of the folder, and in **Create in**, specify where you want the new folder to reside in the hierarchy, and then click **OK**.

   Note: You can create up to 128 folders.

To delete a custom report

1. In the left pane of the Reporting tool, next to **Custom Reports**, click the **Click to manage custom reports** icon.
2. Select the check box that corresponds with the report you want to delete, and then click **Delete**.

   Note: When you delete a folder, all the contents of that folder are deleted.

Modifying the Time Interval

By default, built-in reports display data for the last day. However, if you want to change the time interval for a built-in report, you can save the report as a custom report. The new interval applies to all charts in the report. The following table describes the time-interval options.

Table 1. Time Intervals

Time interval	Displays
 Last Hour	Statistics data collected for the last hour.
 Last Day	Statistics data collected for the last day (24 hours).
 Last Week	Statistics data collected for the last week (7 days).
 Last Month	Statistics data collected for the last month (31 days).
 Last Year	Statistics data collected for the last year (365 days).
 Custom	Statistics data collected for a time period that you are prompted to specify.

To modify the time interval

1. In the left pane of the Reporting tool, click a report.
2. On the report toolbar, click **Duration**, and then click a time interval.

Setting the Data Source and Time Zone

You can retrieve data from different data sources to display them in the reports. For information about data sources, see How Data Collection Works. You can also define the

time zone for the reports and apply the currently displayed report's time selection to all the reports, including the built-in reports.

To set the data source and time zone

1. In the **Reporting tool**, on the report toolbar, click **Settings**.
2. In the **Settings** dialog box, in **Data Source**, select the data source from which you want to retrieve the counter information.
3. Do one or both of the following:
   - If you want the tool to remember the time period for which a chart is plotted, select the **Remember time selection for charts** check box.
   - If you want the reports to use the time settings of your NetScaler appliance, select the **Use Appliance's time zone** check box.

Exporting and Importing Custom Reports

You can share reports with other NetScaler administrators by exporting reports. You can also import reports.

To export or import custom reports

1. In the left pane of the Reporting tool, next to **Custom Reports**, click the **Click to manage custom reports** icon.
2. Select the check box that corresponds with the report you want to export or import, and then click **Export** or **Import**.

   Note: When you export the file, it is exported in a .gz file format.

Working with Charts

Use charts to plot and monitor counters or groups of counters. You can include up to four charts in one report. In each chart, you can plot up to 32 counters. The charts can use different graphical formats (for example, area and bar). You can move the charts up or down within the report, customize the colors and visual display for each counter in a chart, and delete a chart when you do not want to monitor it.

In all report charts, the horizontal axis represents time and the vertical axis represents the value of the counter.

Adding a Chart

When you add a chart to a report, the **System Overview** chart appears with the **CPU Usage** counter plotted for the last one day. To plot a different group of statistics or select a different counter, see Modifying a Chart.

Note: If you add charts to a built-in report, and you want to retain the report, you must save the report as a custom report.

Use the following procedure to add a chart to a report.

To add a chart to a report

1. In the left pane of the **Reporting tool**, click a report.
2. Under the chart where you want to add the new chart, click the **Add** icon.

Modifying a Chart

You can modify a chart by changing the functional group for which the statistics are displayed and by selecting different counters.

To modify a chart

1. In the left pane of the Reporting tool, click a report.
2. Under the chart that you want to modify, click **Counters**.
3. In the dialog box that appears, in the **Title** box, type a name for the chart.
4. Next to **Plot chart for**, do one of the following:
   - To plot counters for global counters, such as Integrated Cache and Compression, click **System global statistics**.
   - To plot entity counters for entity types, such as Load Balancing and GSLB, click **System entities statistics**.
5. In **Select group**, click the desired entity.
6. Under **Counters**, in **Available**, click the counter name(s) that you want to plot, and then click the **>** button.
7. If you selected **System entities statistics** in step 4, on the **Entities** tab, under **Available**, click the entity instance name(s) you want to plot, and then click the **>** button.
8. Click **OK**.

Viewing a Chart

You can specify the graphical formats of the plotted counters in a chart. Charts can be viewed as line charts, spline charts, step-line charts, scatter charts, area charts, bar charts, stacked area charts, and stacked bar charts. You can also zoom in, zoom out, or scroll inside the plot area of a chart. You can zoom in or out for all data sources for 1 hour, 1 day, 1 week, 1 month, 1 year, and 3 years.

Other options for customizing the view of a chart include customizing the axes of the charts, changing the background and edge color of the plot area, customizing the color and size of the grids, and customizing the display of each data set (counter) in a chart.

Data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if **CPU usage** and **Memory usage** are displayed in first and second order at the bottom of the chart, **CPU usage** is equal to **Data Set 1** and **Memory usage** is equal to **Data Set 2**.

Whenever you modify a built-in report, you need to save the report as a custom report to retain your changes.

To change the graph type of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart you want to view, on the chart toolbar, click **Customize**.
3. On the **Chart** tab, under **Category**, click **Plot type**, and then click the graph type you want to display for the chart. If you want to display the graph is 3D, select the **Use 3D** check box.

To refocus a chart with detailed data

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Zoom In**, and do one or both of the following:
   - To refocus the chart to display data for a specific time window, drag and drop the cursor from the start time to the end time. For example, you can view data for a one-hour period on a certain day.
   - To refocus the chart to display data for a data point, simply click once on chart where you want to zoom in and get more detailed information.
3. Once you have the desired range of time for which you want to view detailed data, on the report toolbar, click **Tabular View**. Tabular view displays the data in numeric form in rows and columns.

To view numeric data for a graph

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Tabular View**. To return to the graphical view, click **Graphical View**.

   Note: You can also view the numeric data in the graphical view by hovering your cursor over the notches in the gridlines.

To scroll through time in a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Scroll**, and then click inside the chart and drag the cursor in the direction for which you want to see data for a new time period. For example, if you want to view data in the past, click and drag to the left.

To change the background color and text color of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click **Customize**.
3. On the **Chart** tab, under **Category**, click one or more of the following:
   - To change the background color, click **Background Color**, and then select the options for color, transparency, and effects.
   - To change the text color, click **Text Color**, and then select the options for color, transparency, and effects.

To customize the axes of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click **Customize**.
3. On the **Chart** tab, under **Category**, click one or more of the following:
   - To change the scale of the left y-axis, click **Left Y-Axis**, and then select the scale you want.
   - To change the scale of the right y-axis, click **Right Y-Axis**, in **Data set to plot**, select the date set, and then select the scale you want.

     Note: The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if **CPU usage** and **Memory usage** are displayed in first and second order at the bottom of the chart, **CPU usage** is equal to **Data Set 1** and **Memory usage** is equal to **Data Set 2**.
   - To plot each data set in its own hidden y-axis, click **Multiple Axes**, and then click **Enable**.

To change the background color, edge color, and gridlines for a plot area of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the plot area, click **Customize**.
3. On the **Plot Area** tab, under **Category**, click one or more of the following:
   - To change the background color and edge color of the chart, click **Background Color** and **Edge Color**, and then select the options for color, transparency, and effects.
   - To change the horizontal or vertical grids of the chart, click **Horizontal Grids** or **Vertical Grids**, and then select the options for displaying the grids, grid width, grid color, transparency, and effects.

To change the color and graph type of a data set

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the display of the data set (counters), click **Customize**.
3. On the **Data Set** tab, in **Select Data Set**, select the data set (counter) for which you want to customize the graphical display.

   Note: The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if **CPU usage** and **Memory usage** are displayed in first and second order at the bottom of the chart, **CPU usage** is equal to **Data Set 1** and **Memory usage** is equal to **Data Set 2**.
4. Under **Category**, do one of more of the following:
   - To change the background color, click **Color**, and then select the options for color, transparency, and effects.
   - To change the graph type, click **Plot type**, and then select the graph type you want to display for the data set. If you want to display the graph as 3D, select the **Use 3D** check box.

Exporting Chart Data to Excel

For further data analysis, you can export charts to Excel in a comma-separated value (CSV) format.

To export chart data to Excel

1. In the left pane of the Reporting tool, select a report.

2. In the right pane, under the chart with the data you want to export to Excel, click **Export**.

Deleting a Chart

If you do not want to use a chart, you can remove it from the report. You can permanently remove charts from custom reports only. If you delete a chart from a built-in report and want to retain the changes, you need to save the report as a custom report.

To delete a chart

1. In the left pane of the Reporting tool, select a report.

2. In the right pane, under the chart that you want to delete, click the **Delete** icon.

Examples

To display the trend report for CPU usage and memory usage for the last week

1. In the left pane of the Reporting tool, under **Built-in Reports**, expand **System**.
2. Click the report **CPU vs. Memory Usage and HTTP Requests Rate**.
3. In the right pane, on the report toolbar, click **Duration**, and then click **Last Week**.

To compare the bytes received rate and the bytes transmitted rate between two interfaces for the last week

1. In the right pane, on the report toolbar, click **Create.**
2. In the **Report Name** box, type a name for the custom report (for example, `Custom_Interfaces`), and then click **OK**. The report is created with the default **System Overview** chart, which displays the **CPU Usage** counter plotted for the last hour.
3. Under **System Overview**, on the chart toolbar, click **Counters**.
4. In the counter selection pane, in **Title**, type a name for the chart (for example, `Interfaces bytes data`).
5. In **Plot chart for**, click **System entities statistics**, and then in **Select Group**, select **Interface**.
6. On the **Entities** tab, click the interface name(s) you want to plot (for example, **1/1** and **1/2**), and then click the **>** button.
7. On the **Counters** tab, click **Bytes received (Rate)** and **Bytes transmitted (Rate)** and then click the **>** button.
8. Click **OK**.
9. On the report toolbar, click **Duration**, and then click **Last Week**.

Stopping and Starting the Data Collection Utility

The performance data is stored in different data sources on the Citrix® NetScaler® appliance. The default data source is /var/log/db/default. You can create up to 32 data sources.

The data collection utility nscollect retrieves data from the NetScaler and updates the data source. This utility runs automatically when you start the NetScaler. It creates a database for global counters at /var/log/db/<DataSourceName>. The entity-specific databases are created based on the entities configured on the NetScaler. A specific folder is created for each entity type in

/var/log/db/<DataSourceName/EntityNameDB>

Before creating a database for an entity, nscollect allocates a unique number to the entity and creates the database based on that number. It retrieves all the counters available for a group. However, there is a limit on the number of different entities that nscollect can retrieve, as described in the following table.

Table 1. Limits on Entity Numbers Retrieved by nscollect

Entity name	Limit
Content Switching Virtual Servers	100
Cache Redirection Virtual Servers	50
DOS Policies	100
GSLB Domains	100
GSLB Services	100
GSLB Sites	32
GSLB Virtual Servers	100
Interfaces	8
LB Virtual Servers	100
ACLs	100
ACL6	50
Priority Queuing Policies	100
RNAT IP Addresses	100
SureConnect Policies	100
Services	250
Service Groups	100

System CPU	8
VLAN	25
VPN Virtual Servers	5

The nscollect utility retrieves n number of entity counters and creates the entity database. If the first *n* counters change in the subsequent fetch, the database stores more than n entries for that entity type. However, you need to delete the unused entity counters manually.

Note: The Reporting tool supports only numerical counters.

By default, nscollect retrieves data at every 5-minute interval. Data is maintained in 5-minute granularity for one day, hourly for the last 30 days, and daily for three years.

When you start the NetScaler, the nscollect utility automatically starts. However, if data is not updated accurately, or there is corrupted data displayed in the reports, you can stop and then restart the utility. You may also want to stop nscollect to back up the databases or to create a new data source.

To stop nscollect

At a NetScaler command prompt, type the following:

/netscaler/nscollect stop
You can start nscollect on either the local system or a remote system.

To start nscollect on the local system

At a NetScaler command prompt, type the following:

/netscaler/nscollect start

To start nscollect on the remote system

At a NetScaler command prompt, type the following:

/netscaler/nscollect start **-U** *NS_IP:UserName:Password* **-ds** *DataSourceName*

Example

/netscaler/nscollect start -U 10.102.29.170:nsroot:nsroot -ds default

Advanced Networking

The following topics provide a conceptual reference and instructions for configuring the various networking components on the NetScaler appliance.

IP Addressing	Learn the various types of NetScaler-owned IP addresses and how to create, customize, and remove them.
Interfaces	Configure some of the basic network configurations that must be done to get started.
Access Control Lists (ACLs)	Configure the different types of Access Control Lists and how to create, customize, and remove them.
IP Routing	Learn and configure the routing functionality of the NetScaler appliance, both static and dynamic.
Internet Protocol version 6 (IPv6)	Learn how the NetScaler appliance supports IPv6.
High Availability	Learn how High Availability (HA) works in a NetScaler deployment to ensure uninterrupted operation in any transaction.

IP Addressing

Before you can configure the NetScaler® appliance, you must assign the NetScaler IP Address (NSIP), also known as the Management IP address. You can also create other NetScaler-owned IP addresses for abstracting servers and establishing connections with the servers. In this type of configuration, the appliance serves as a proxy for the abstracted servers. You can also proxy connections by using network address translations (INAT and RNAT). When proxying connections, the appliance can behave either as a bridging (Layer 2) device or as a packet forwarding (Layer 3) device. To make packet forwarding more efficient, you can configure static ARP entries. For IPv6, you can configure neighbor discovery (ND).

IP Addressing

Before you can configure the NetScaler® appliance, you must assign the NetScaler IP Address (NSIP), also known as the Management IP address. You can also create other NetScaler-owned IP addresses for abstracting servers and establishing connections with the servers. In this type of configuration, the appliance serves as a proxy for the abstracted servers. You can also proxy connections by using network address translations (INAT and RNAT). When proxying connections, the appliance can behave either as a bridging (Layer 2) device or as a packet forwarding (Layer 3) device. To make packet forwarding more efficient, you can configure static ARP entries. For IPv6, you can configure neighbor discovery (ND).

Configuring NetScaler-Owned IP Addresses

The NetScaler-owned IP Addresses—NetScaler IP Address (NSIP), Virtual IP Addresses (VIPs), Subnet IP Addresses (SNIPs), Mapped IP Addresses (MIPs), and Global Server Load Balancing Site IP Addresses (GSLBIPs)—exist only on the NetScaler appliance. The NSIP uniquely identifies the NetScaler on your network, and it provides access to the appliance. A VIP is a public IP address to which a client sends requests. The NetScaler terminates the client connection at the VIP and initiates a connection with a server. This new connection uses a SNIP or a MIP as the source IP address for packets forwarded to the server. If you have multiple data centers that are geographically distributed, each data center can be identified by a unique GSLBIP.

You can configure some NetScaler-owned IP addresses to provide access for management applications.

Configuring the NetScaler IP Address (NSIP)

The NetScaler IP (NSIP) address is the IP address at which you access the NetScaler for management purposes. The NetScaler can have only one NSIP, which is also called the Management IP address. You must add this IP address when you configure the NetScaler for the first time. If you modify this address, you must reboot the NetScaler. You cannot remove an NSIP address. For security reasons, NSIP should be a non-routable IP address on your organization's LAN.

Note: Configuring the NetScaler IP address is mandatory.

To create the NetScaler IP address by using the NetScaler command line

At the NetScaler command prompt, type:

- set ns config [**-IPAddress** <ip_addr> **-netmask** <netmask>]
- show ns config

Example

```
> set ns config -ipaddress 10.102.29.170 -netmask 255.255.255.0
 Done
> show ns config
        NetScaler IP: 10.102.29.170  (mask: 255.255.255.0)
        Number of MappedIP(s): 1
        Node: Standalone
        Global configuration settings:
                        HTTP port(s): (none)
                    Max connections: 0
          Max requests per connection: 0
                 Client IP insertion: DISABLED
                     Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                        Min Path MTU: 576
               Path MTU entry timeout: 10
                     FTP Port Range: 0
                          Timezone: GMT-11:00-SST-Pacific/Pago_Pago
 Done
```

Parameters for configuring the NSIP address

IPAddress

Unique identification used to represent an entity. This is a mandatory parameter.

netmask

Subnet mask associated with the IP address. This is a mandatory parameter.

type

Type of the IP address. Possible values: SNIP, VIP, MIP, and GSLBsiteIP.

To configure the NetScaler IP address by using the configuration utility

1. In the navigation pane, click **NetScaler**.
2. On the **System Overview** page, click **Setup Wizard**.
3. In the **Setup Wizard** dialog box, click **Next**.
4. Under **System Configuration**, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring the NetScaler IP address" as shown:
   - **IP Address***—IPAddress
   - **Netmask***—netmask

   * A required parameter
5. Follow the instructions in the **Setup Wizard** to complete the configuration.

Configuring and Managing Virtual IP Addresses (VIPs)

Configuration of a virtual server IP address (VIP) is not mandatory during initial configuration of the NetScaler. When you configure load balancing, you assign VIPs to virtual servers.

For more information about configuring the load balancing setup, see the "Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

In some situations, you need to customize VIP attributes or enable or disable a VIP. A VIP is usually associated with a virtual server, and some of the attributes of the VIP are customized to meet the requirements of the virtual server. You can host the same virtual server on multiple NetScaler appliances residing on the same broadcast domain, by using ARP and ICMP attributes. After you add a VIP (or any IP address), the NetScaler sends, and then responds to, ARP requests. VIPs are the only NetScaler-owned IP addresses that can be disabled. When a VIP is disabled, the virtual server using it goes down and does not respond to ARP, ICMP, or L4 service requests.

As an alternative to creating VIPs one at a time, you can specify a consecutive range of VIPs.

To create a VIP address by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns ip <IPAddress> <netmask> **-type** <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.59 255.255.255.0 -type VIP
 Done
> show ns ip 10.102.29.59

      IP: 10.102.29.59
      Netmask: 255.255.255.0
      Type: VIP
      state: Enabled
```

```
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Enabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
Warning: management access is disabled
Done
```

To create a range of VIP addresses by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns ip <IPAddress> <netmask> **-type** <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
ip "10.102.29.60" added
ip "10.102.29.61" added
ip "10.102.29.62" added
ip "10.102.29.63" added
ip "10.102.29.64" added
 Done

> show ip
  Ipaddress       Type          Mode    Arp     Icmp    Vserver  State
  ---------       ----          ----    ---     ----    -------  ------
1) 10.102.29.170   NetScaler IP    Active   Enabled  Enabled  NA      Enabled
2) 10.102.29.171   MIP           Active   Enabled  Enabled  NA      Enabled
.
.
46) 10.102.29.60    VIP           Active   Enabled  Enabled  Enabled  Enabled
47) 10.102.29.61    VIP           Active   Enabled  Enabled  Enabled  Enabled
48) 10.102.29.62    VIP           Active   Enabled  Enabled  Enabled  Enabled
49) 10.102.29.63    VIP           Active   Enabled  Enabled  Enabled  Enabled
50) 10.102.29.64    VIP           Active   Enabled  Enabled  Enabled  Enabled
 Done
```

Parameters for configuring VIP addresses

ipAddress (IP Address)

Unique identification used to represent an entity. This is a required parameter.

netmask (Netmask)

Subnet mask associated with the IP address. This is a required parameter.

type (Type)

Type of the IP address. Specify **VIP**.

arp (ARP)

Use Address Resolution Protocol (ARP) to map IP addresses to the corresponding hardware addresses. Possible values: Enabled, Disabled. Default: Enabled.

icmpresponse (ICMP Response)

NetScaler sends ICMP responses to PING requests according to this value. The user network applications that use ICMP are PING and TRACEROUTE. This parameter can be set only if type is set as VIP. Possible values: NONE, ONE_VSERVER, ALL_VSERVERS, and VSVR_CNTRLD. Default value: NONE.

- When you select NONE, NetScaler always responds (even when the virtual server is DOWN).
- When you select ONE_VSERVER, NetScaler responds if at least one virtual server on this IP address is UP.
- When you select ALL_VSERVERS, NetScaler responds only if all the virtual servers on this IP address are UP.
- When you select VSVR_CNTRLD, the behavior depends on the ICMP VSERVER RESPONSE setting on the virtual server.

The following settings can be made on a virtual server:

- When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
- When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

Note: This parameter is supported only on NetScaler 9.3.e.

arpresponse (ARP Response)

NetScaler appliance sends ARP responses according to this value. This parameter can be set only if type is set as VIP. Possible values: NONE, ONE_VSERVER. Default value: NONE.

- When you select NONE, NetScaler always responds (even when the virtual server is DOWN).
- When you select ONE_VSERVER, NetScaler responds if at least one virtual server on this IP address is UP.
- When you select ALL_VSERVERS, NetScaler responds only if all the virtual servers on this IP address are UP.

Note: This parameter is supported only on NetScaler 9.3.e.

vServer (Virtual Server)

Apply the vserver attribute to this IP address. Possible values: Enabled, Disabled. Default: Enabled.

state (State)

State of the VIP. Possible values: Enabled, Disabled. Default: Enabled.

To configure a VIP address by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, do one of the following:
   - To create a new IP, click **Add**.
   - To modify an existing IP, select the IP, and then click **Open**.
3. In the **Create IP** or **Configure IP** dialog box, set the following parameters:
   - **IP Address***
   - **Netmask***
   - **IP Type**: Select **VIP**.
   - **ARP Response**
   - **ICMP Response**
   - **ARP**
   - **Virtual Server**
   - **Dynamic Routing**
   - **Host Route**
   - **Gateway IP***
   - **Metric**
   - **V Server RHI Level**
   - **OSPF LSA Type**
   - **Area**

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. The IP address that you configured appears in the details pane.

To create a range of VIP addresses by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, click **Add Range**.
3. In the **Create IP - Range** dialog box, set the following parameters:
   - **IP Address***
   - **Netmask***
   - **Type**—type. Select **VIP**.
   - **IP Type**
   - **ARP**
   - **ICMP Response**
   - **Virtual Server**
   - **Dynamic Routing**
   - **Host Route**
   - **Gateway IP***
   - **Metric**
   - **V Server RHI Level**
   - **OSPF LSA Type**
   - **Area**

   *A required parameter
4. Click **Create**, and then click **Close**. The range of IP addresses that you created appears in the details pane.

To enable or disable an IPv4 VIP address by using the NetScaler command line

At the NetScaler command prompt, type one of the following sets of commands to enable or disable a VIP and verify the configuration:

- enable ns ip <IPAddress>
- show ns ip <IPAddress>

- disable ns ip <IPAddress>
- show ns ip <IPAddress>

Example

```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Enabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Disabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled

 Done
```

To enable or disable a VIP address by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, on the **IPv4s** tab, select the VIP address and do one of the following:
   - To enable the selected IP address, click **Enable**.
   - To disable the selected IP address, click **Disable**.
3. In the details pane, verify that the VIP address is enabled or disabled, as appropriate.

Configuring ARP response Suppression for Virtual IP addresses (VIPs)

You can configure the NetScaler appliance to respond or not respond to ARP requests for a Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP.

Note: This feature is supported only on NetScaler 9.3.e.

For example, if virtual servers V1, of type HTTP, and V2, of type HTTPs, share VIP address 10.102.29.45 on a NetScaler appliance, you can configure the appliance to not respond to any ARP request for VIP 10.102.29.45 if both V1 and V2 are in the DOWN state.

The following three options are available for configuring ARP-response suppression for a virtual IP address.

- **NONE**. The NetScaler appliance responds to any ARP request for the VIP address, irrespective of the state of the virtual servers associated with the address.
- **ONE VSERVER**. The NetScaler appliance responds to any ARP request for the VIP address if at least one of the associated virtual servers is in UP state.
- **ALL VSERVER**. The NetScaler appliance responds to any ARP request for the VIP address if all of the associated virtual servers are in UP state.

Following table shows the sample behavior of NetScaler appliance for a VIP configured with two virtual servers:

Associated virtual servers for a VIP	**STATE 1**	**STATE 2**	**STATE 3**	**STATE 4**
NONE				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	Yes	Yes	Yes
ONE VSERVER				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	Yes	Yes	No
ALL VSERVER				

V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	No	No	No

Consider an example where you want to test the performance of two virtual servers, V1 and V2, which have the same VIP address but are of different types and are each configured on NetScaler appliances NS1 and NS2. Let's call the shared VIP address *VIP1*.

V1 load balances servers S1, S2, and S3. V2 load balances servers S4 and S5.

On both NS1 and NS2, for VIP1, the ARP suppression parameter is set to ALL_VSERVER. If you want to test the performance of V1 and V2 on NS1, you must manually disable V1 and V2 on NS2, so that NS2 does not respond to any ARP request for VIP1.

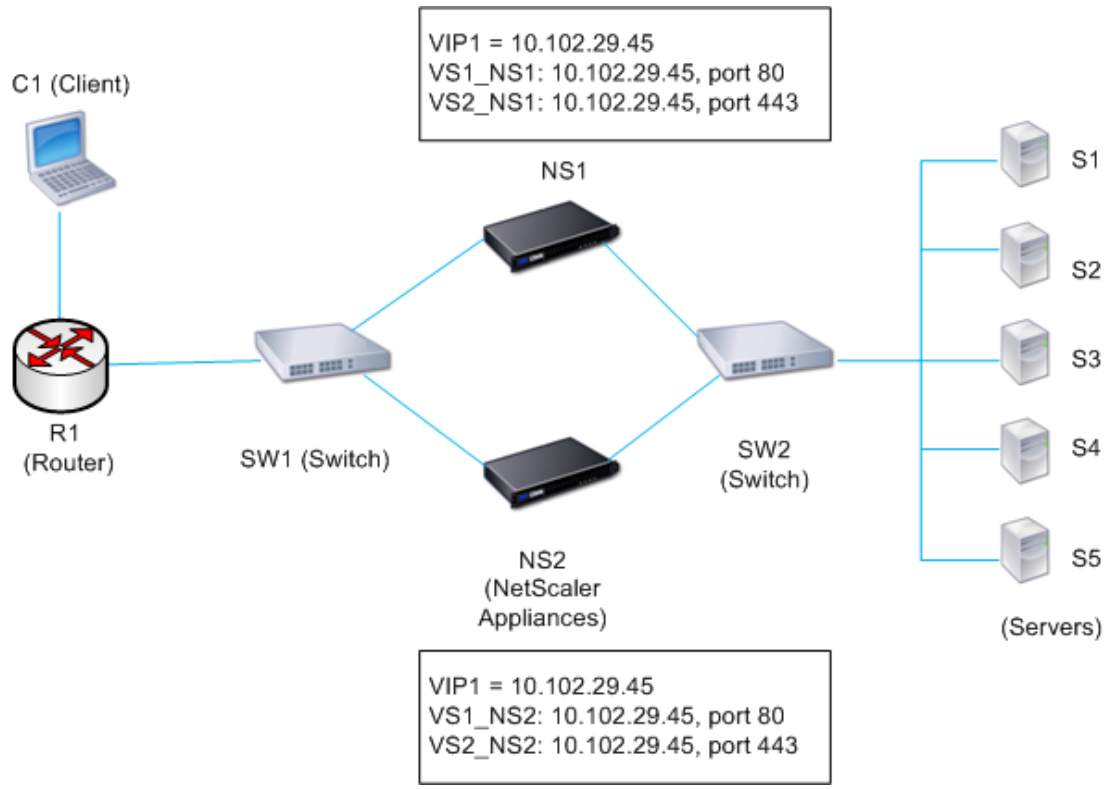


Figure 1.

The execution flow is as follows:

1. Client C1 sends a request to V1. The request reaches R1.
2. R1 does not have an APR entry for the IP address (VIP1) of V1, so R1 broadcasts an ARP request for VIP1.
3. NS1 replies with source MAC address MAC1 and source IP address VIP1. NS2 does not reply to the ARP request.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table, and R1 updates the ARP entry with MAC1 and VIP1.

5. R1 forwards the packet to address VIP1 on NS1.

6. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP or MIP addresses and S2. When S2 sends a response to the client, the response returns by the same path.

7. Now you want to test the performance of V1 and V2 on NS2, so you enable V1 and V2 on NS2 and disable them on NS1. NS2 now broadcasts an ARP message for VIP1. In the message, MAC2 is the source MAC address and VIP1 is the source IP address.

8. SW1 learns the port number for reaching MAC2 from the ARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS2. R1 updates its ARP table.

9. Now suppose the ARP entry for VIP1 times out in the ARP table of R1, and client C1 sends a request for V1. Because R1 does not have an APR entry for VIP1, it broadcasts an ARP request for VIP1.

10. NS2 replies with a source MAC address and VIP1 as the source IP address. NS1 does not reply to the ARP request.

To configure ARP response supression by using the NetScaler command line

At the NetScaler command prompt, type:

- set ns ip -arpResponse <arpResponse>]
- show ns ip <IPAddress>

Example

```
> set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
 Done
> show ns ip 10.102.29.96

      IP: 10.102.29.96
      Netmask: 255.255.255.255
      Type: VIP
      state: Enabled
      arp: Enabled
      arpResponse: ALL_VSERVERS
                        icmp: Enabled
      icmpResponse: NONE
                        vserver: Enabled
      management access: Disabled
        telnet: Disabled
        ftp: Disabled
        ssh: Disabled
```

```
        gui: Disabled
        snmp: Enabled
      Restrict access: Disabled
      dynamic routing: Disabled
      hostroute: Disabled
Warning: management access is disabled
Done
```

Parameter for configuring ARP response supression

arpresponse (ARP Response)

NetScaler appliance sends ARP responses according to this value. This parameter can be set only if type is set as VIP. Possible values: NONE, ONE_VSERVER. Default value: NONE.

- When you select NONE, NetScaler always responds (even when the virtual server is DOWN).
- When you select ONE_VSERVER, NetScaler responds if at least one virtual server on this IP address is UP.
- When you select ALL_VSERVERS, NetScaler responds only if all the virtual servers on this IP address are UP.

To configure ARP response supression by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, select the IP, and then click **Open**.
3. In the **Configure IP** dialog box, set the following parameter:
   - **ARP Response**

   *A required parameter
4. Click **OK**, and then click **Close**.

Configuring Subnet IP Addresses (SNIPs)

A subnet IP (SNIP) address is used in connection management and server monitoring. It is not mandatory to specify a SNIP when you initially configure the NetScaler appliance. In a multiple-subnet scenario, the NetScaler IP (NSIP) address, the mapped IP (MIP) address, and the IP address of a server can exist on different subnets. To eliminate the need to configure additional routes on devices such as servers, you can configure subnet IP addresses (SNIPs) on the NetScaler. With Use SNIP (USNIP) mode enabled, a SNIP is the source IP address of a packet sent from the NetScaler to the server, and the SNIP is the IP address that the server uses to access the NetScaler. This mode is enabled by default.

When you add a SNIP, a route corresponding to the SNIP is added to the routing table. The NetScaler determines the next hop for a service from the routing table, and if the IP address of the hop is within the range of a SNIP, the NetScaler uses the SNIP to source traffic to the service. When multiple SNIPs cover the IP addresses of the next hops, the SNIPs are used in round robin manner.

The following figure illustrates USNIP mode.

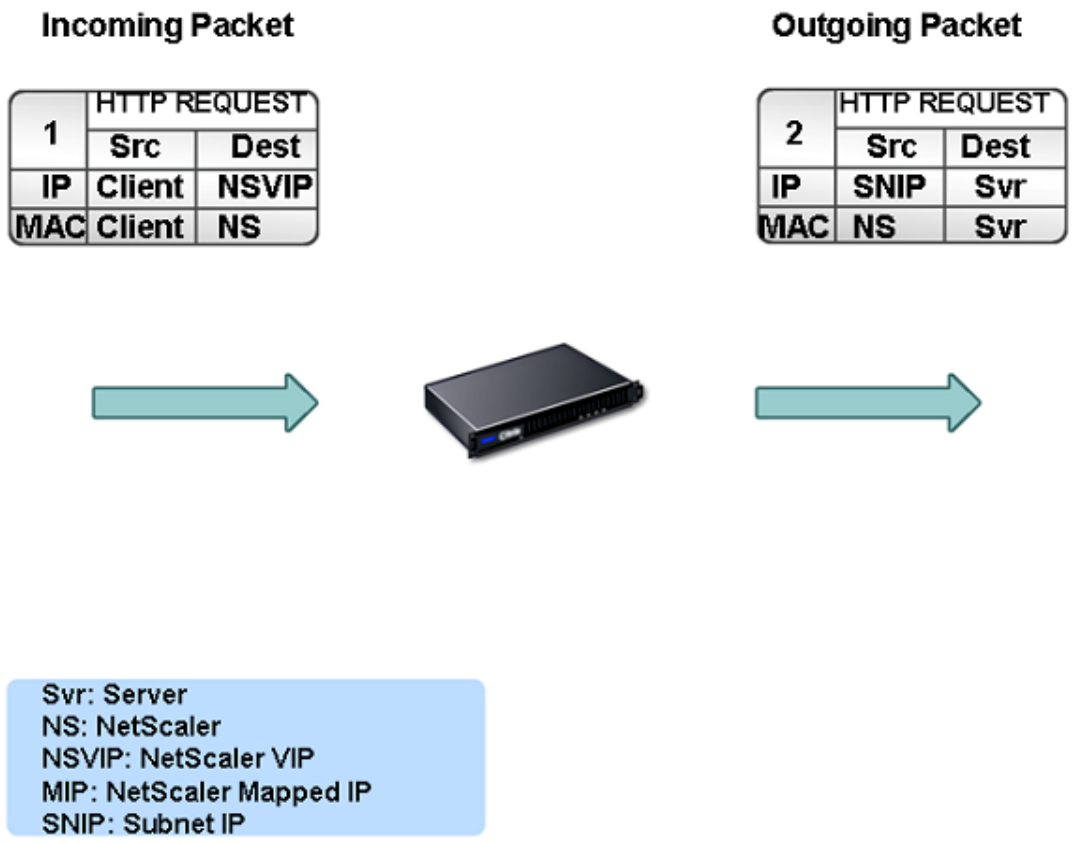


Figure 1. USNIP Mode

As an alternative to creating SNIPs one at a time, you can specify a consecutive range of SNIPs.

To configure a SNIP address by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns ip <IPAddress> <netmask> **-type** <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.203 255.255.255.0 -type SNIP
 Done
> sh ns ip 10.102.29.103

   IP: 10.102.29.103
   Netmask: 255.255.255.0
   Type: SNIP
   state: Enabled
   arp: Enabled
   icmp: Enabled
   vserver: NA
   management access: Disabled
     telnet: Enabled
     ftp: Enabled
     ssh: Enabled
     gui: Enabled
     snmp: Enabled
   Restrict access: Disabled
   dynamic routing: Disabled
   hostroute: Disabled
   # free ports: 1032111
 Warning: management access is disabled
 Done
```

To create a range of SNIP addresses by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns ip <IPAddress> <netmask> **-type** <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done

> sh ns ip
  Ipaddress       Type          Mode     Arp      Icmp     Vserver  State
  ---------       ----          ----     ---      ----     -------  ------
1) 10.102.29.170   NetScaler IP    Active   Enabled  Enabled  NA      Enabled
2) 10.102.29.171   MIP           Active   Enabled  Enabled  NA      Enabled
.
.
51) 10.102.29.205   SNIP          Active   Enabled  Enabled  NA      Enabled
52) 10.102.29.206   SNIP          Active   Enabled  Enabled  NA      Enabled
53) 10.102.29.207   SNIP          Active   Enabled  Enabled  NA      Enabled
54) 10.102.29.208   SNIP          Active   Enabled  Enabled  NA      Enabled
55) 10.102.29.209   SNIP          Active   Enabled  Enabled  NA      Enabled
 Done
```

Parameters for configuring SNIP addresses

IPAddress

Unique identification used to represent an entity. This is a required parameter.

netmask

Subnet mask associated with the IP address. This is a required parameter.

type

Type of the IP address. Specify **SNIP**.

To configure a SNIP address by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, do one of the following:
   - To create a new IP address, click **Add**.
   - To modify an existing IP address, select the address, and then click **Open**.
3. In the **Create IP** or **Configure IP** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring SNIP addresses" as shown:
   - **IP Address***—IPAddress
   - **Netmask***—netmask
   - **Type**—type (Select **SNIP**.)

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. The IP address that you configured appears in the details pane.

To create a range of SNIP addresses by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, click **Add Range**.
3. In the **Create IP - Range** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring SNIP addresses" as shown:
   - **IP Address***—IPAddress
   - **Netmask***—netmask
   - **Type**—type (Select **SNIP**.)

   *A required parameter
4. Click **Create**, and then click **Close**. The range of IP addresses that you created appears in the details pane.

To enable or disable USNIP mode by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable ns mode usnip
- disable ns mode usnip

To enable or disable USNIP mode by using the configuration utility

1. In the navigation pane, expand **System** and click **Settings**.
2. In the details pane, in the **Modes and Features** group, click **Change modes**.
3. In the **Configure Modes** dialog box, do one of the following:
   - To enable USNIP, select the **Use Subnet IP** check box.
   - To disable USNIP, clear the **Use Subnet IP** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Configuring Mapped IP Addresses (MIPs)

Mapped IP addresses (MIP) are used for server-side connections. A MIP can be considered a default Subnet IP (SNIP) address, because MIPs are used when a SNIP is not available or Use SNIP (USNIP) mode is disabled.

If the mapped IP address is the first in the subnet, the NetScaler appliance adds a route entry, with this IP address as the gateway to reach the subnet. You can create or delete a MIP during run time without rebooting the appliance.

As an alternative to creating MIPs one at a time, you can specify a consecutive range of MIPs.

To create a MIP address by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns ip <IPAddress> <netmask> **-type** <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.171 255.255.255.0 -type MIP
 Done
> sh ns ip 10.102.29.171

   IP: 10.102.29.171
   Netmask: 255.255.255.0
   Type: MIP
   state: Enabled
   arp: Enabled
   icmp: Enabled
   vserver: NA
   management access: Disabled
     telnet: Enabled
     ftp: Enabled
     ssh: Enabled
     gui: Enabled
     snmp: Enabled
   Restrict access: Disabled
   dynamic routing: Disabled
   hostroute: Disabled
```

```
   # free ports: 1031960
Warning: management access is disabled
Done
```

To create a range of MIP addresses by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns ip <IPAddress> <netmask> **-type** <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.[173-175] 255.255.255.0 -type MIP
ip "10.102.29.173" added
ip "10.102.29.174" added
ip "10.102.29.175" added
Done

> sh ns ip
  Ipaddress       Type          Mode     Arp     Icmp    Vserver  State
  ---------       ----          ----     ---     ----    -------  ------
1) 10.102.29.170   NetScaler IP    Active   Enabled  Enabled  NA      Enabled
2) 10.102.29.171   MIP           Active   Enabled  Enabled  NA      Enabled
.
.
56) 10.102.29.173   MIP          Active   Enabled  Enabled  NA      Enabled
57) 10.102.29.174   MIP          Active   Enabled  Enabled  NA      Enabled
58) 10.102.29.175   MIP          Active   Enabled  Enabled  NA      Enabled
 Done
```

Parameters for configuring MIP addresses

IPAddress

Unique identification used to represent an entity. This is a required parameter.

netmask

Subnet mask associated with the IP address. This is a required parameter.

type

Type of the IP address. Specify **MIP**.

To configure a MIP address by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, do one of the following:
   - To create a new IP address, click **Add**.
   - To modify an existing IP address, select the address, and then click **Open**.
3. In the **Create IP** or **Configure IP** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring MIP addresses” as shown:
   - **IP Address***—IPAddress
   - **Netmask***—netmask
   - **Type**—type (Select **MIP**.)

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. The IP address that you configured appears in the details pane.

To create a range of MIP addresses by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, click **Add Range**.
3. In the **Create IP - Range** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring MIP addresses” as shown:
   - **IP Address***—IPAddress
   - **Netmask***—netmask
   - **Type**—type (Select **MIP**.)

   *A required parameter
4. Click **Create**, and then click **Close**. The range of IP addresses that you created appears in the details pane.

Configuring GSLB Site IP Addresses (GSLBIP)

A GSLB site IP (GSLBIP) address is an IP address associated with a GSLB site. It is not mandatory to specify a GSLBIP address when you initially configure the NetScaler appliance. A GSLBIP address is used only when you create a GSLB site.

For more information about creating a GSLB site IP address, see the "Global Server Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Removing a NetScaler-Owned IP Address

You can remove any IP address except the NSIP. The following table provides information about the processes you must follow to remove the various types of IP addresses. Before removing a VIP, remove the associated virtual server.

Table 1. Implications of Removing a NetScaler-Owned IP Address

IP address type	Implications
Subnet IP address (SNIP)	If IP address being removed is the last IP address in the subnet, the associated route is deleted from the route table. If the IP address being removed is the gateway in the corresponding route entry, the gateway for that subnet route is changed to another NetScaler-owned IP address.
Mapped IP address (MIP)	If a SNIP exists, you can remove the MIPs. The NetScaler uses NSIP and SNIPs to communicate with the servers when the MIP is removed. Therefore, you must also enable use SNIP (USNIP) mode. For information about enabling and disabling USNIP mode, see Configuring Subnet IP Addresses (SNIPs).
Virtual Server IP address (VIP)	Before removing a VIP, you must first remove the vserver associated with it. For information about removing the vserver, see the "Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.
GSLB-Site-IP address	Before removing a GSLB site IP address, you must remove the site associated with it. For information about removing the site, see the "Global Server Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To remove an IP address by using the NetScaler command line

At the NetScaler command prompt, type:

rm ns ip <IPaddress>

Example

rm ns ip 10.102.29.54

To remove an IP address by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. On the **IPs** page, on the **IPv4s** tab, select the IP address that you want to remove, and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**. A message appears in the status bar, stating that the IP address has been removed successfully.

Configuring Application Access Controls

Application access controls, also known as management access controls, form a unified mechanism for managing user authentication and implementing rules that determine user access to applications and data. You can configure MIPs and SNIPs to provide access for management applications. Management access for the NSIP is enabled by default and cannot be disabled. You can, however, control it by using ACLs.

For information about using ACLs, see Access Control Lists (ACLs).

The NetScaler appliance does not support management access to VIPs.

The following table provides a summary of the interaction between management access and specific service settings for Telnet.

Management Access	Telnet (State Configured on the NetScaler)	Telnet (Effective State at the IP Level)
Enable	Enable	Enable
Enable	Disable	Disable
Disable	Enable	Disable
Disable	Disable	Disable

The following table provides an overview of the IP addresses used as source IP addresses in outbound traffic.

Application/ IP	NSIP	MIP	SNIP	VIP
ARP	Yes	Yes	Yes	No
Server side traffic	No	Yes	Yes	No
RNAT	No	Yes	Yes	Yes
ICMP PING	Yes	Yes	Yes	No
Dynamic routing	Yes	No	Yes	Yes

The following table provides an overview of the applications available on these IP addresses.

Application/ IP	NSIP	MIP	SNIP	VIP
SNMP	Yes	Yes	Yes	No
System access	Yes	Yes	Yes	No

You can access and manage the NetScaler by using applications such as Telnet, SSH, GUI, and FTP.

Note: Telnet and FTP are disabled on the NetScaler for security reasons. To enable them, contact the customer support. After the applications are enabled, you can apply the controls at the IP level.

To configure the NetScaler to respond to these applications, you need to enable the specific management applications. If you disable management access for an IP address, existing connections that use the IP address are not terminated, but no new connections can be initiated.

Also, the non-management applications running on the underlying FreeBSD operating system are open to protocol attacks, and these applications do not take advantage of the NetScaler appliance's attack prevention capabilities.

You can block access to these non-management applications on a MIP, SNIP, or NSIP. When access is blocked, a user connecting to a NetScaler by using the MIP, SNIP, or NSIP is not be able to access the non-management applications running on the underlying operating system.

To configure management access for an IP address by using the NetScaler command line

At the NetScaler command prompt, type:

set ns ip <IPAddress> **-mgmtAccess** <value> **-telnet** <value> **-ftp** <value> **-gui** <value> **-ssh** <value> **-snmp** <value> **-restrictAccess** (ENABLED | DISABLED)

Example

```
set ns ip 10.102.29.54 -mgmtAccess enabled -restrictAccess ENABLED
```

Parameters for customizing a SNIP or MIP address

telnet

Allow Telnet access to the IP address. Possible values: ENABLED, DISABLED. Default: ENABLED.

ftp

Allow File Transfer Protocol (FTP) access to the IP address. Possible values: ENABLED, DISABLED. Default: ENABLED.

gui

Allow Graphical User Interface (GUI) access to the IP address. Possible values: ENABLED, SECUREONLY, DISABLED. Default: ENABLED.

ssh

Allow Secure Shell (SSH) access to the IP address. Possible values: ENABLED, DISABLED. Default: ENABLED.

snmp

Allow Simple Network Management Protocol (SNMP) access to the IP address. Possible values: ENABLED, DISABLED. Default: ENABLED.

mgmtAccess

Allow external access to the IP address. Possible values: ENABLED, DISABLED. Default: DISABLED.

dynamicRouting

Allow dynamic routing on the IP address. Specific to SNIP. Possible values: Enabled, Disabled. Default: Disabled.

restrictAccess

Block access to non-management applications on this IP. This options is applicable for MIPs, SNIPs, and NSIP, and is disabled by default. Non-management applications may run on the underlying NetScaler Free BSD operating system. Possible values: ENABLED, DISABLED. Default: DISABLED.

To enable management access for an IP address by using the configuration utility

1. In the navigation pane, expand **Network** and click **IPs**.
2. On the **IPs** page, select the IP address that you want to modify (for example, **10.102.29.54**), and then click **Open**.
3. In the **Configure IP** dialog box, under **Application Access Control**, select the **Enable Management Access control to support the below listed applications** check box.
4. Select the application or applications that you want to enable.
5. To block access to non-management applications on an IP address, select the **Allow access only to management applications** check box.
6. Click **OK**.

How the NetScaler Proxies Connections

When a client initiates a connection, the NetScaler appliance terminates the client connection, initiates a connection to an appropriate server, and sends the packet to the server. The appliance does not perform this action for service type UDP or ANY.

For more information about service types, see "Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

You can configure the NetScaler to process the packet before initiating the connection with a server. The default behavior is to change the source and destination IP addresses of a packet before sending the packet to the server. You can configure the NetScaler to retain the source IP address of the packets by enabling Use Source IP mode.

How the Destination IP Address Is Selected

Traffic sent to the NetScaler appliance can be sent to a virtual server or to a service. The appliance handles traffic to virtual servers and services differently. The NetScaler terminates traffic received at a virtual server IP (VIP) address and changes the destination IP address to the IP address of the server before forwarding the traffic to the server, as shown in the following diagram.

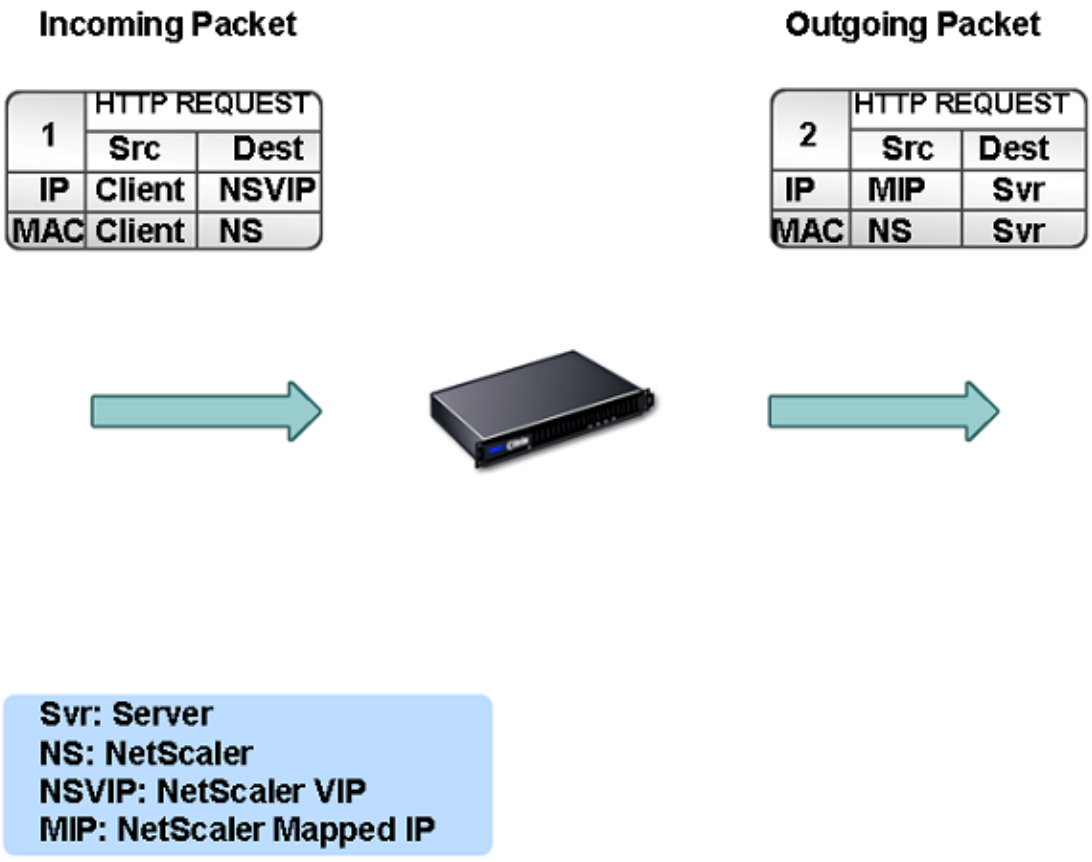


Figure 1. Proxying Connections to VIPs

Packets destined for a service are sent directly to the appropriate server, and the NetScaler does not modify the destination IP addresses. In this case, the NetScaler functions as a proxy.

How the Source IP Address Is Selected

When the NetScaler appliance communicates with the physical servers or peer devices, by default, it does not use the IP address of the client. NetScaler maintains a pool of mapped IP addresses (MIPs) and subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address of a connection to the physical server. Depending on the subnet in which the physical server is placed, NetScaler decides whether a MIP should be used or SNIP.

Note: If the Use Source IP (USIP) option is enabled, NetScaler uses the IP address of the client.

Enabling Use Source IP Mode

When the NetScaler appliance communicates with the physical servers or peer devices, by default, it uses one of its own IP addresses as the source IP. The appliance maintains a pool of mapped IP addresses (MIPs) and subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address for a connection to the physical server. The decision of whether to select a MIP or a SNIP depends on the subnet in which the physical server resides.

If necessary, you can configure the NetScaler appliance to use the client's IP address as source IP. Some applications need the actual IP address of the client. The following use cases are a few examples:

- Client's IP address in the web access log is used for billing purposes or usage analysis.
- Client's IP address is used to determine the country of origin of the client or the originating ISP of the client. For example, many search engines such as Goggle provide content relevant to the location to which the user belongs.
- The application must know the client's IP address to verify that the request is from a trustworthy source.
- Sometimes, even though an application server does not need the client's IP address, a firewall placed between the application server and the NetScaler may need the client's IP address for filtering the traffic.

Enable **Use Source IP** mode (USIP) mode if you want NetScaler to use the client's IP address for communication with the servers. By default, USIP mode is disabled. USIP mode can be enabled globally on the NetScaler or on a specific service. If you enable it globally, USIP is enabled by default for all subsequently created services. If you enable USIP for a specific service, the client's IP address is used only for the traffic directed to that service.

As an alternative to USIP mode, you have the option of inserting the client's IP address (CIP) in the request header of the server-side connection for an application server that needs the client's IP address.

In earlier NetScaler releases, USIP mode had the following source-port options for server-side connections:

- Use the client's port. With this option, connections cannot be reused. For every request from the client, a new connection is made with the physical server.
- Use proxy port. With this option, connection reuse is possible for all requests from the same client. Before NetScaler release 8.1 this option imposed a limit of 64000 concurrent connections for all server-side connections.

In the later NetScaler releases , if USIP is enabled, the default is to use a proxy port for server-side connections and not reuse connections. Not reusing connections may not effect the speed of establshing connections.

By default, the **Use Proxy Port** option is enabled if the **USIP** mode is enabled. For more information about the Use Proxy Port option, see Using the Client Port When Connecting to the Server.

Note: f you enable the **USIP** mode, it is recommended to enable the **Use Proxy Port** option.

The following figureshows how the NetScaler uses IP addresses in USIP mode.

Figure 1. IP Addressing in USIP Mode

Recommended Usage

Enable **USIP** in the following situations:

- Load balancing of Intrusion Detection System (IDS) servers
- Stateless connection failover
- Sessionless load balancing
- If you use the Direct Server Return (DSR) mode

Note: When USIP is required in the one-arm mode installation of the NetScaler appliance, make sure that the server's gateway is one of the IP addresses owned by the NetScaler. For more information about NetScaler owned IP addresses, see Configuring NetScaler owned IP addresses.

- If you enable USIP, set the idle timeout for server connections to a value lower than the default value, so that idle connections are cleared quickly on the server side.

  For more information about setting an idle time-out value, see "Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

- For transparent cache redirection, if you enable USIP, enable **L2CONN** also.

- Because HTTP connections are not reused when USIP is enabled, a large number of server-side connections may accumulate. Idle server connections can block connections for other clients. Therefore, set limits on maximum number of connections to a service. Citrix also recommendeds setting the HTTP server time-out value, for a service on which USIP is enabled, to a value lower than the default, so that idle connections are cleared quickly on the server side.

To globally enable or disable USIP mode by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable ns mode usip
- disable ns mode usip

To enable USIP mode for a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**usip** (YES | NO)

Example

set service Service-HTTP-1 -usip YES

To globally enable or disable USIP mode by using the configuration utility

1. In the navigation pane, expand **System** and click **Settings**.
2. On the **Settings** page, under **Modes and Features**, click **Configure modes**.
3. In the **Configure Modes** dialog box, do one of the following:
   - To enable Use Source IP mode, select the **Use Source IP** check box.
   - To disable Use Source IP mode, clear the **Use Source IP** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

To enable USIP mode for a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to enable the USIP mode, and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Settings**, select the **Use Source IP** check box.
5. Click **OK**.

Configuring Network Address Translation

Network address translation (NAT) involves modification of the source and/or destination IP addresses and/or the TCP/UDP port numbers of IP packets that pass through the NetScaler appliance. Enabling NAT on the appliance enhances the security of your private network, and protects it from a public network such as the Internet, by modifying your network's source IP addresses when data passes through the NetScaler. Also, with the help of NAT entries, your entire private network can be represented by a few shared public IP addresses. The NetScaler supports the following two types of network address translation:

- Inbound NAT (INAT), in which the NetScaler replaces the destination IP address in the packets generated by the client with the private IP address of the server.
- Reverse NAT (RNAT), in which the NetScaler replaces the source IP address in the packets generated by the servers with the public NAT IP addresses.

Configuring INAT

When a client sends a packet to a NetScaler appliance that is configured for Inbound Network Address Translation (INAT), the appliance translates the packet's public destination IP address to a private destination IP address and forwards the packet to the server at that address.

The following configurations are supported:

- **IPv4-IPv4 Mapping**: A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.
- **IPv4-IPv6 Mapping**: A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance creates an IPv6 request packet with the IP address of the IPv6 server as the destination IP address.
- **IPv6-IPv4 Mapping**: A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance creates an IPv4 request packet with the IP address of the IPv4 server as the destination IP address.
- **IPv6-IPv6 Mapping**: A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.

When the appliance forwards a packet to a server, the source IP address assigned to the packet is determined as follows:

- If use subnet IP (USNIP) mode is enabled and use source IP (USIP) mode is disabled, the NetScaler uses a subnet IP address (SNIP) as the source IP address.
- If USNIP mode is disabled and USIP mode is disabled, the NetScaler uses a mapped IP address (MIP) as the source IP address.
- If USIP mode is enabled, and USNIP mode is disabled the NetScaler uses the client IP (CIP) address as the source IP address.
- If both USIP and USNIP modes are enabled, USIP mode takes precedence.
- You can also configure the NetScaler to use a unique IP address as the source IP address, by setting the proxyIP parameter.
- If none of the above modes is enabled and a unique IP address has not been specified, the NetScaler attempts to use a MIP as the source IP address.

- If both USIP and USNIP modes are enabled and a unique IP address has been specified, the order of precedence is as follows: USIP-unique IP-USNIP-MIP-Error.

To protect the NetScaler from DoS attacks, you can enable TCP proxy. However, if other protection mechanisms are used in your network, you may want to disable them.

You can create, modify, or remove an INAT entry.

To create an INAT entry by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an INAT entry and verify its configuration:

- add inat <name> <publicIP> <privateIP> [**-tcpproxy** (ENABLED | DISABLED)] [**-ftp** (ENABLED | DISABLED)] [**-usip** (ON | OFF)] [**-usnip** (ON | OFF)] [**-proxyIP** <ip_addr|ipv6_addr>]

- show inat [<name>]

Example

```
> add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
 Done
> show inat ip4-ip4

1) NAME: ip4-ip4
   Public IP: 172.16.1.2
   Private IP: 192.168.1.1
   Tcpproxy: DISABLED
   Ftp:       DISABLED
   USNIP : ON
   USIP: ON
   Proxy IP: 10.102.29.171
 Done
```

To modify an INAT entry by using the NetScaler command line

To modify an INAT entry, type the set inat command, the name of the entry, and the parameters to be changed, with their new values.

To remove an INAT configuration by using the NetScaler command line

At the NetScaler command prompt, type:

rm inat <name>

Example

```
> rm inat ip4-ip4
 Done
```

Basic parameters for configuring INAT

name

Name of the Inbound NAT entry being added.

publicIP

Public destination IP address of packets received on the NetScaler. This IP address can be an IPv4 or IPv6 address. Possible values: NetScaler-owned VIPs.

privateIP

Private destination IP address of the server to which the packet is sent by the NetScaler. This IP address can be an IPv4 or IPv6 address. Possible values: IP addresses of the servers.

usip

Use source IP mode. Possible values: Enabled, Disabled. Default: Enabled.

usnip

Use subnet IP mode. Possible values: Enabled, Disabled. Default: Enabled.

proxyIP

A unique IP address used as the source IP address in packets sent to the server.

tcpproxy

Allow TCP traffic. Possible values: Enabled, Disabled. Default: Disabled.

ftp

Allow FTP. Possible values: Enabled, Disabled. Default: Disabled.

To configure an INAT entry by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Routes**.
2. On the **Routes** page, on the **INAT** tab, do one of the following:
   - To create a new INAT entry, click **Add**.
   - To modify an existing INAT entry, select the entry, and then click **Open**.
3. In the **Create INAT** or **Configure INAT** dialog box, specify values for the following parameters, which correspond to parameters described in "Basic parameters for configuring INAT" as shown:
   - **Name*** —name
   - **Public IP Address*** —publicIP

     Note: To use an IPv6 address, select the **IPv6** check box and enter the address in IPv6 format.
   - **Private IP Address*** —privateIP

     Note: To use an IPv6 address, select the **IPv6** check box and enter the address in IPv6 format.
   - **Proxy IP Address** —proxyIP
   - **TCP Proxy Mode** —tcpproxy
   - **FTP Mode** —ftp
   - **Use Source IP Mode** —usip
   - **Use Subnet IP Mode** —usnip

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the INAT entry has been configured successfully.

To remove an INAT configuration by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Routes**.
2. On the **INAT** tab, select the name of the INAT configuration that you want to remove.
3. Click **Remove**, and then click **Close**. A message appears in the status bar, stating that the INAT has been removed successfully.

Coexistence of INAT and Virtual Servers

If both INAT and RNAT are configured, the INAT rule takes precedence over the RNAT rule. If RNAT is configured with a network address translation IP (NAT IP) address, the NAT IP address is selected as the source IP address for that RNAT client.

The default public destination IP in an INAT configuration is the virtual IP (VIP) address of the NetScaler device. virtual servers also use VIPs. When both INAT and a virtual server use the same IP address, the Vserver configuration overrides the INAT configuration.

Following are a few sample configuration setup scenarios and their effects.

Case	Result
You have configured a virtual server and a service to send all data packets received on a specific NetScaler port to the server directly. You have also configured INAT and enabled TCP. Configuring INAT in this manner sends all data packets received through a TCP engine before sending them to the server.	All packets received on the NetScaler, except those received on the specified port, pass through the TCP engine.
You have configured a virtual server and a service to send all data packets of service type TCP, that are received on a specific port on the NetScaler, to the server after passing through the TCP engine. You have also configured INAT and disabled TCP. Configuring INAT in this manner sends the data packets received directly to the server.	Only packets received on the specified port pass through the TCP engine.
You have configured a virtual server and a service to send all data packets received to either of two servers. You are attempting to configure INAT to send all data packets received to a different server.	The INAT configuration is not allowed.
You have configured INAT to send all received data packets directly to a server. You are attempting to configure a virtual server and a service to send all data packets received to two different servers.	The vserver configuration is not allowed.

Configuring RNAT

In Reverse Network Address Translation (RNAT), the NetScaler appliance replaces the source IP addresses in the packets generated by the servers with public NAT IP addresses. By default, the appliance uses a Mapped IP address (MIP) as the NAT IP address. You can also configure the appliance to use a unique NAT IP address for each subnet. You can also configure RNAT by using Access Control Lists (ACLs).

Use Source IP (USIP), Use Subnet IP (USNIP), and Link Load Balancing (LLB) modes affect the operation of RNAT. You can display statistics to monitor RNAT.

You can use either a network addrress or an extended ACL as the condition for an RNAT entry:

- **Using a Network address.** When you use a network address, RNAT processing is performed on all of the packets coming from the specified network.
- **Using Extended ACLs.**When you use ACLs, RNAT processing is performed on all packets that match the ACLs. To configure the NetScaler appliance to use a unique IP address for traffic that matches an ACL, you must perform the following three tasks:

  1. Configure the ACL.
  2. Configure RNAT to change the source IP address and Destination Port.
  3. Apply the ACL.

  The following diagram illustrates RNAT configured with an ACL.

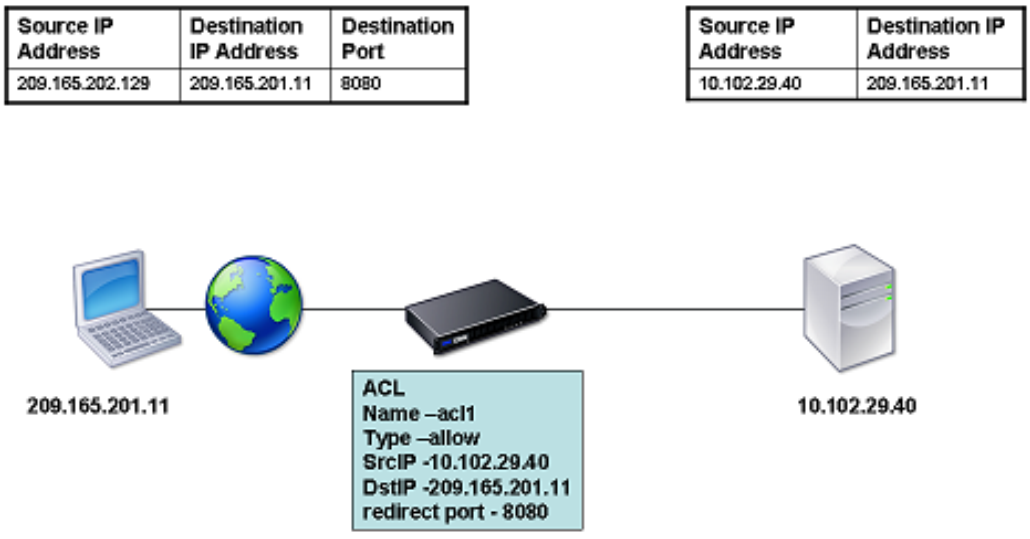


Figure 1. RNAT with an ACL

You have the following basic choices for the type of NAT IP address:

- **Using a MIP or SNIP as the NAT IP Address.** When using a MIP as the NAT IP address, the NetScaler appliance replaces the source IP addresses of server-generated packets with the a MIP. Therefore, the MIP address must be a public IP address. If Use Subnet IP (USNIP) mode is enabled, the NetScaler can use a subnet IP address (SNIP) as the NAT IP address.

- **Using a Unique IP Address as the NAT IP Address.** When using a unique IP address as the NAT IP address, the NetScaler appliance replaces the source IP addresses of server-generated packets with the unique IP address specified. The unique IP address must be a public NetScaler-owned IP address. If multiple NAT IP addresses are configured for a subnet, NAT IP selection uses the round robin algorithm.

  This configuration is illustrated in the following diagram.

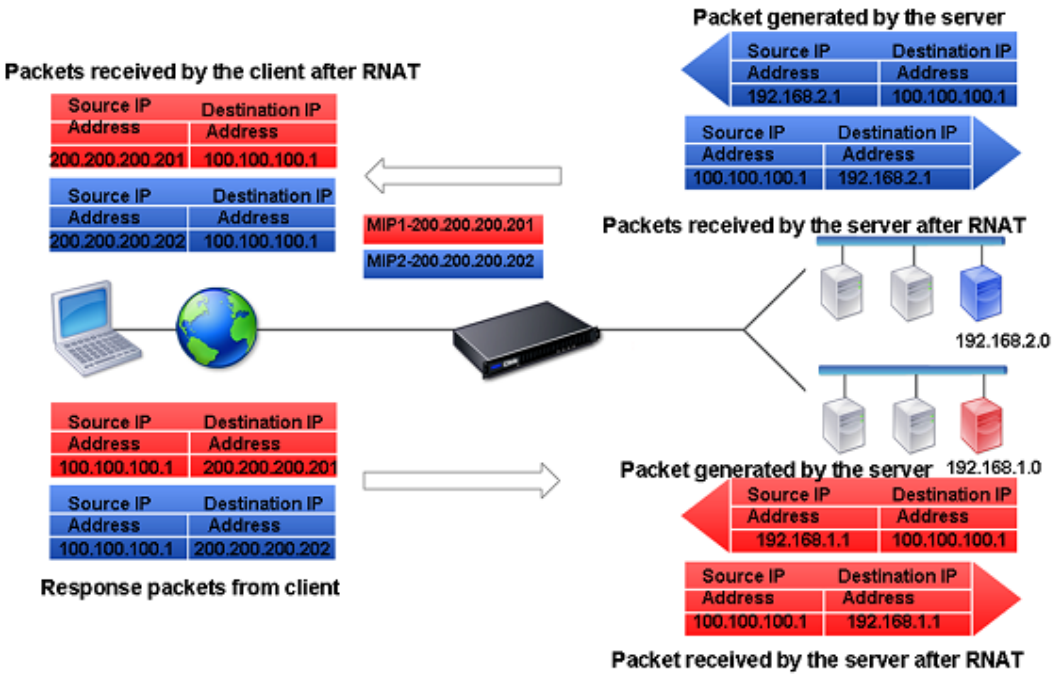


Figure 2. Using a Unique IP Address as the NAT IP Address

Creating an RNAT Entry

The following instructions provide separate command-line procedures for creating RNAT entries that use different conditions and different types of NAT IP addresses. In the configuration utility, all of the variations can be configured in the same dialog box, so there is only one procedure for configuration utility users.

To create an RNAT entry by using the NetScaler command line

At the NetScaler command prompt, type one the following commands to create, respectively, an RNAT entry that uses a network address as the condition and a MIP or SNIP as the NAT IP address, an RNAT entry that uses a network address as the condition and a unique IP address as the NAT IP address, an RNAT entry that uses an ACL as the condition and a MIP or SNIP as the NAT IP address, or an RNAT entry that uses an ACL as a condition and a unique IP address as the NAT IP address:

- set rnat <IPAddress> <netmask>
- set rnat IPAddress <netMask> **-natip** <NATIPAddress>
- set rnat <aclname> [**-redirectPort** <port>]
- set rnat <aclname> [**-redirectPort** <port>] **-natIP** <NATIPAddress>

  Use the following command to verify the configuration:

- show rnat

Examples

A network address as the condition and a MIP or SNIP as the NAT IP address:

```
> set rnat 192.168.1.0 255.255.255.0
 Done
> show rnat
1)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: *
 Done
```

A network address as the condition and a unique IP address as the NAT IP address:

```
> set rnat 192.168.1.0 255.255.255.0 -natip 10.102.29.50
 Done
```

```
> show rnat
1)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.50
 Done
```

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-ow

```
> set rnat 192.168.1.0 255.255.255.0  -natIP 10.102.29.[50-110]
 Done
> show rnat
1)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.59
2)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.66
3)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.67
4)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.79
5)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.90
6)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.102
7)      Network: 192.168.1.0      Netmask: 255.255.255.0
        NatIP: 10.102.29.103
 Done
```

An ACL as the condition and a MIP or SNIP as the NAT IP address:

```
> set rnat acl1
 Done
> show rnat
1)      ACL Name: acl1
        NatIP: *
 Done
```

An ACL as a condition and a unique IP address as the NAT IP address:

```
> set rnat acl1  -natIP 209.165.202.129
Done
> show rnat
1)      ACL Name: acl1
        NatIP: 209.165.202.129
 Done
```

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-ow

```
> set rnat acl1  -natIP 10.102.29.[50-70]
 Done
> show rnat
1)      ACL Name: acl1
        NatIP: 10.102.29.59
2)      ACL Name: acl1
        NatIP: 10.102.29.66
3)      ACL Name: acl1
        NatIP: 10.102.29.67
```

Done

Parameters for creating an RNAT entry

IPAddress

Address of the network or subnet from which the traffic is flowing.

netmask

Subnet mask associated with the network.

aclname

The name of an extended ACL. The rule of the ACL will be used as an RNAT rule.

redirectPort

The redirect port.

natip

Any NetScaler-owned IPv4 address except the NSIP address. The NetScaler appliance replaces the source IP addresses of server-generated packets with the IP address specified. The IP address must be a public NetScaler-owned IP address. If you specify multiple NetScaler-owned IP addresses for this field, NAT IP selection uses the round robin algorithm for each session. At the NetScaler command line, you can specify a range of IP addresses for this field. All the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified will be set for this field.

To create an RNAT entry by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Routes**.
2. On the **Routes** page, click the **RNAT** tab.
3. In the details pane, click **Configure RNAT**.
4. In the **Configure RNAT** dialog box, do one of the following:
   - If you want to use the network address as a condition for creating an RNAT entry, click **Network**. Specify values for the following parameters, which correspond to parameters described in "Parameters for creating an RNAT entry" as shown:
     - **Network**—IPAddress
     - **Netmask**—netmask
   - If you want to use an extended ACL as a condition for creating an RNAT entry, click **ACL**. Specify values for the following parameters, which correspond to parameters described in "Parameters for creating an RNAT entry" as shown:
     - **ACL Name**—aclname
     - **Redirect Port**—redirectPort
5. To set a MIP or SNIP as a NAT IP, jump to Step 7.
6. To set a unique IP address as a NAT IP, in the **Available NAT IP (s)** list, select the IP address that you want to set as the NAT IP, and then click **Add**. The NAT IP you selected appears in the **Configured NAT IP(s)** list.
7. Click **Create**, and then **Close**. A message appears in the status bar, stating that the RNAT has been configured successfully.

Monitoring RNAT

You can display RNAT statistics to troubleshoot issues related to IP address translation.

To view RNAT statistics by using the NetScaler command line

At the NetScaler command prompt, type:

stat rnat

Example

```
> stat rnat

RNAT summary
                        Rate (/s)          Total
Bytes Received                0                0
Bytes Sent                    0                0
Packets Received              0                0
Packets Sent                  0                0
Syn Sent                      0                0
Current RNAT sessions         --               0
 Done
>
```

The following tables describes the statistics associated with RNAT and RNAT IP.

Table 1. RNAT Statistics

Statistic	Description
Bytes received	Bytes received during RNAT sessions
Bytes sent	Bytes sent during RNAT sessions
Packets received	Packets received during RNAT sessions
Packets sent	Packets sent during RNAT sessions
Syn sent	Requests for connections sent during RNAT sessions

Current sessions	Currently active RNAT sessions

To monitor RNAT by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Routes**.
2. In the details pane, on the **RNAT** tab, click **Statistics**. The **Statistics** dialog box appears, displaying the RNAT statistics.

RNAT in USIP, USNIP, and LLB Modes

When RNAT and Use Source IP (USIP) are both configured, RNAT takes precedence. When RNAT and USNIP are configured, selection of the source IP address is based on the state of USNIP, as follows:

- If USNIP is off, the NetScaler appliance uses the mapped IP addresses.
- If USNIP is on, the NetScaler uses a SNIP as the NAT IP address.

This behavior does not apply when a unique NAT IP address is used.

In a topology where the NetScaler appliance performs both Link Load Balancing (LLB) and RNAT for traffic originating from the server, the appliance selects the source IP address based on the router. The LLB configuration determines selection of the router.

For more information about LLB, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Configuring Prefix-Based IPv6-IPv4 Translation

Prefix-based translation is a process of translating packets sent from private IPv6 servers into IPv4 packets, using an IPv6 prefix configured in the NetScaler appliance. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix.

The NetScaler appliance compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix. If there is a match, the NetScaler appliance generates an IPv4 packet and sets the destination IP address as the last 32 bits of the destination IP address of the matched IPv6 packet. IPv6 packets addressed to this prefix have to be routed to the NetScaler so that the IPv6-IPv4 translation is done by the NetScaler.

In the following diagram, 3ffe::/96 is configured as the IPv6 NAT prefix on NetScaler NS1. The IPv6 host sends an IPv6 packet with destination IP address 3ffe::74.125.91.105. NS1 compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix, and they match. NS1 then generates an IPv4 packet and sets the destination IP address as 74.125.91.105.

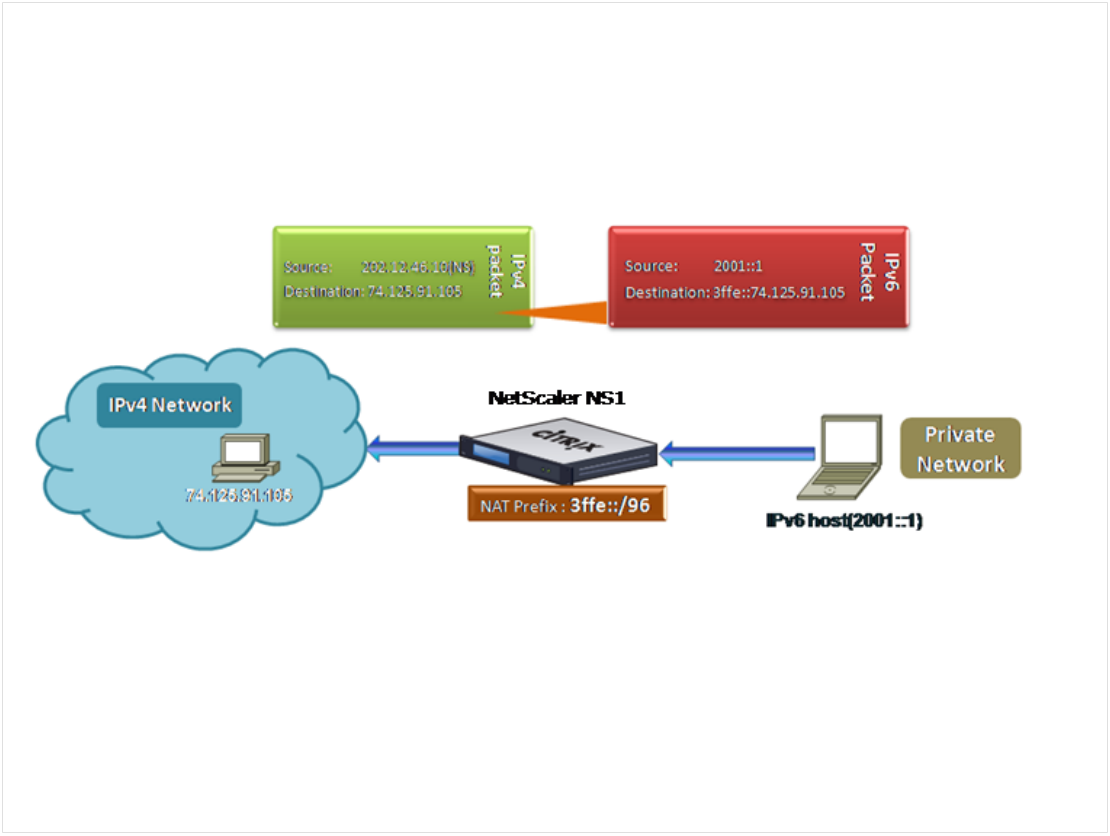


Figure 1. IPv6-IPv4 Prefix-Based Translation

To configure prefix-based IPv6-IPv4 translation by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set a nat prefix and verify its configuration:

- set ipv6 [**-natprefix** <ipv6_addr|*>]
- show ipv6

Example

```
> set ipv6 -natprefix 3ffe::/96
 Done
> show ipv6
 IPv6 NAT prefix : 3ffe::/96
 IPv6 RA learning : DISABLED
 ND6 base reachable time : 30000 ms
 ND6 computed reachable time : 16848 ms
 ND6 retransmission time : 1000 ms
 Done
```

Parameter for configuring prefix-based IPv6-IPv4 translation

natprefix

The prefix used for translating packets from private IPv6 servers to IPv4 packets. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix. IPv6 packets addressed to this prefix have to be routed to the NetScaler so that the IPv6-IPv4 translation is done by the NetScaler.

To configure prefix-based IPv6-IPv4 translation by using the configuration utility

1. In the navigation pane, expand **Network**.
2. In the details pane, in the **Settings** group, click **Change IPv6 Settings**.
3. In the **Configure IPv6 settings** dialog box, set the following parameter, which corresponds to the parameter described in "Parameter for configuring prefix-based IPv6-IPv4 translation" as shown:
   - **IPv6 NAT prefix**—natprefix
4. Click **OK**. A message appears in the status bar, stating that the IPv6 NAT prefix entry has been configured successfully.

Configuring Static ARP

You can add static ARP entries to and remove static ARP entries from the ARP table. After adding an entry, you should verify the configuration. If the IP address, port, or MAC address changes after you create a static ARP entry, you must remove or manually adjust the static entry. Therefore, creating static ARP entries is not recommended unless necessary.

To add a static ARP entry by using the NetScaler command line

At the NetScaler command prompt, type:

- add arp **-IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name>
- show arp <IPAddress>

Example

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
> show arp 10.102.29.6
      IP             MAC              Iface VLAN  Origin
      --             ---              ----- ----  ------
1)     10.102.29.6      00:24:e8:73:ca:ec  1/1   1     DYNAMIC
 Done
```

To remove a static ARP entry by using the NetScaler command line

At the NetScaler command prompt, type the rm arp command and the IP address.

Parameters for adding a static ARP entry

IPAddress

The IP address of the server.

mac

The MAC address of the server. Type the MAC address with colons (:) as shown in the example above.

ifnum

The physical interface for the ARP entry. Use the show interface command to view the valid interface names.

To add a static ARP entry by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **ARP Table**.
2. On the **ARP Table** page, in the details pane, click **Add**.
3. In the **Create ARP entry** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for adding an ARP entry" as shown:
   - **IP Address***—ip
   - **MAC Address***—mac
   - **Interface Number***—ifnum

   *A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the ARP entry has been configured successfully.

Setting the Timeout for Dynamic ARP Entries

You can globally set an aging time (time-out value) for dynamically learned ARP entries. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time.

You can specify an ARP time-out value of from 1 through 1200 seconds.

To set the time-out for dynamic ARP entries by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the time-out for dynamic ARP entries and verify its configuration:

- set arpparam **-timeout** <positive_integer>]
- show arpparam

Example

```
> set arpparam -timeout 500
 Done
> show arpparam
       ARP Parameters
      Aging time for ARP table entry : 500
 Done
```

To set the time-out for dynamic ARP entries to its default value by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the time-out for dynamic ARP entries to its default value and verify its configuration:

- unset arpparam
- show arpparam

Example

```
> unset arpparam
 Done
> show arpparam
       ARP Parameters
       Aging time for ARP table entry : 1200
 Done
```

To set the time-out for dynamic ARP entries by using the configuration utility

1. In the navigation pane, click **Network**.
2. In the details pane, in the **Settings** group, click **Configure ARP Global Parameters**.
3. In the **Configure ARP Global Parameters** dialog box, type a value for **ARP Table Entry Timeout**.
4. Click **OK**. A message appears in the status bar, stating that the ARP Global settings have been changed successfully.

Configuring Neighbor Discovery

Neighbor discovery (ND) is one of the most important protocols of IPv6. It is a message-based protocol that combines the functionality of the Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and Router Discovery. ND allows nodes to advertise their link layer addresses and obtain the MAC addresses or link layer addresses of the neighboring nodes.This process is performed by the Neighbor Discovery protocol (ND6).

Neighbor discovery can perform the following functions:

Router Discovery

Enables a host to discover the local routers on an attached link and automatically configure a default router.

Prefix Discovery

Enables the host to discover the network prefixes for local destinations.

Note: Currently, the NetScaler does not support Prefix Discovery.

Parameter Discovery

Enables a host to discover additional operating parameters, such as MTU and the default hop limit for outbound traffic.

Address Autoconfiguration

Enables hosts to automatically configure IP addresses for interfaces both with and without stateful address configuration services such as DHCPv6. The NetScaler does not support Address Autoconfiguration for Global IPv6 addresses.

Address Resolution

Equivalent to ARP in IPv4, enables a node to resolve a neighboring node's IPv6 address to its link-layer address.

Neighbor Unreachability Detection

Enables a node to determine the reachability state of a neighbor.

Duplicate Address Detection

Enables a node to determine whether an NSIP address is already in use by a neighboring node.

Redirect

Equivalent to the IPv4 ICMP Redirect message, enables a router to redirect the host to a better first-hop IPv6 address to reach a destination.

Note: The NetScaler does not support IPv6 Redirect.

To enable neighbor discovery, you create entries for the neighbors.

Adding IPv6 Neighbors

Adding IPv6 neighbors enables neighbor discovery.

To add an IPv6 neighbor by using the NetScaler command line

At the NetScaler command prompt, type:

- add nd6 <neighbor> <mac> <ifnum> [**-vlan** <integer>]
- show nd6

Example

```
> add nd6 2001::1 00:04:23:be:3c:06 1/1 –vlan 1
 Done
> show nd6
   Neighbor              MAC-Address(Vlan, Interface)      State       TIME
   --------              ----------------------------      -----     --------
1) ::1                   00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE     PERMANENT
2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE     PERMANENT
3) 2001::1               00:04:23:be:3c:06(   1,   1/1)   REACHABLE       STATIC
 Done
```

Neighbor Discovery Parameters

neighbor

IPv6 neighbor entry. Mandatory.

mac

Unique address assigned to identify the network appliance. Mandatory.

ifnum

The interface on which the MAC resides. Mandatory.

vlan

Virtual LAN (VLAN) that the neighbor is part of.

To add an IPv6 neighbor by using the configuration utility

1. In the navigation pane, expand **Network** and click **IPv6 Neighbors**.
2. In the details pane, click **Add**.
3. In the **CreateIPv6 Neighbor** dialog box, in the **Neighbor** and **MAC Address** text boxes, respectively, type IPv6 address and MAC Address of the neighbour (for example, 3ffe:100:100::1, 00:d0:68:0b:58:da).
4. If the neighbor is part of a VLAN, in the and **VLAN** field, type the VLAN ID (for example, 1).
5. In the **Interface** list box, select the interface of the neighbour (for example, LO/1).
6. Click **Create**, and click **Close**.

Removing IPv6 Neighbors

To remove a neighbor discovery entry by using the NetScaler command line

At the NetScaler command prompt, type:

rm nd6 <Neighbour> **-vlan** <VLANID>

Example

```
rm nd6 3ffe:100:100::1 -vlan 1
```

To remove all neighbor discovery entries by using the NetScaler command line

At the NetScaler command prompt, type:

clear nd6

To remove a neighbor discovery entry by using the configuration utility

1. In the navigation pane, expand **Network** and click **IPv6 Neighbor**.
2. In the details pane, select the neighbour entry that you want to remove (for example,3ffe:100:100::1).
3. Click **Remove**.

To remove all neighbor discovery entries by using the configuration utility

1. In the navigation pane, expand **Network** and click **IPv6 Neighbor**.
2. In the **IPv6 Neighbors** page, click **Clear**.

Configuring IP Tunnels

An IP Tunnel is a communication channel, that can be created by using encapsulation technologies, between two networks that do not have a routing path. Every IP packet that is shared between the two networks is encapsulated within another packet and then sent via the tunnel.

The NetScaler appliance implements IP Tunneling in the following ways:

- NetScaler as an Encapsulator (Load Balancing with DSR mode)
- NetScaler as a Decapsulator

NetScaler as an Encapsulator (Load Balancing with DSR Mode)

Consider an organization that has multiple data centers across different countries, where the NetScaler maybe at one location and the back-end servers are located in a different country. In essence, the NetScaler and the back-end servers are on different networks and are connected via a router.

When you configure Direct Server Return (DSR) on this NetScaler, the packet sent from the source subnet is encapsulated by the NetScaler and sent via a router and tunnel to the appropriate back-end server. The back-end server decapsulates the packet and responds directly to the client, without allowing the packet to pass via the NetScaler.

NetScaler as a Decapsulator

Consider an organization having multiple data centers each having NetScalers and back-end servers. When a packet is sent from data center A to data center B it is usually sent via an intermediary, say a router or another NetScaler. The NetScaler processes the packet and then forwards the packet to the back-end server. However, if an encapsulated packet is sent, the NetScaler must be able to decapsulate the packet before sending it to the back-end servers. To enable the NetScaler to function as a decapsulator, a tunnel is added between the router and the NetScaler. When the encapsulated packet, with additional header information, reaches the NetScaler, the data packet is decapsulated i.e. the additional header information is removed, and the packet is then forwarded to the appropriate back-end servers.

The NetScaler can also be used as a decapsulator for the Load Balancing feature, specifically in scenarios when the number of connections on a vserver exceeds a threshold value and all the new connections are then diverted to a back-up vserver.

Creating IP Tunnels

For enabling IP/IP (IP Tunneling) for a specific virtual IP (VIP) address you need to create an IP tunnel manually, known as configured tunnels. For Cloud Bridge, you need to create GRE tunnels.

To create an IP tunnel by using the NetScaler command line

At the NetScaler command prompt type:

- add iptunnel <name> <remoteIp> <remoteSubnetMask> <localIp> **-type** -**protocol** (ipoverip | GRE) -**secure** (YES | NO)
- show iptunnel

To modify or remove an IP tunnel by using the NetScaler command line

- To modify an IP tunnel, type the set iptunnel command, the name of the tunnel, and the parameters to be changed, with their new values.
- To remove an IP tunnel, type the rm iptunnel command and the name of the tunnel.

Parameters for creating an IP tunnel

name

Name of the IP Tunnel. This alphanumeric string is required and cannot be changed after the service group is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

remoteIp

A public IPv4 address of the remote NetScaler appliance used to set up the tunnel.

remoteSubnetMask

Subnet mask of the remote IP address of the tunnel.

localIp

A public IPv4 address of the local NetScaler appliance used to set up the tunnel. Possible values: Auto, MIP, SNIP, and VIP. Default: Auto.

protocol

The protocol to be used in setting up the IP tunnel. Select GRE for using the Generic Routing Encapsulation (GRE) protocol to set up a GRE tunnel.

secure

Enable or disable IPSec for securing communication in the GRE tunnel.

To create an IP Tunnel by using the configuration utility

1. In the navigation pane, expand **Network**, and click **IP Tunnels**.
2. In the details pane, click **Add**.
3. In the **Add IP Tunnel** dialog box, specify values for the following parameters:
   - **Name*—**name
   - **Remote IP*—**remoteIp
   - **Remote Mask*—**remoteSubnetMask
   - **Local IP Type***—localIp (in the **local IP Type** drop down list, select one of the IP type (Mapped IP, Subnet IP, and Virtual). All the configured IPs of the selected IP type will be populated in the **Local IP** drop down list. Select the desired IP from the list.)
   - **Protocol—**protocol
   - **Secure—**secure

   *A required parameter.
4. Click **Create**, and then click **Close.**

Customizing IP Tunnels Globally

By globally specifying the source IP address, you can assign a common source IP address across all tunnels. Also, because fragmentation is CPU-intensive, you can globally specify that the NetScaler appliance drop any packet that requires fragmentation. Alternatively, if you would like to fragment all packets as long as a CPU threshold value is not reached, you can globally specify the CPU threshold value.

To globally customize IP tunnels by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally customize IP tunnels and verify the configuration:

- set iptunnelparam -srcIP <sourceIPAddress> **-srcIPRoundRobin** (**YES** | **NO**)-dropFrag [**YES** | **NO**] **-dropFragCpuThreshold** <Positive integer>
- show iptunnelparam

Example

```
> set iptunnelparam –srcIP 12.12.12.22 -dropFrag Yes –dropFragCpuThreshold 50
 Done
> show iptunnelparam
Tunnel Source IP: 12.12.12.22
Round Robin of Tunnel Source IP: NO
Drop if Fragmentation Needed: Yes
CPU usage threshold to avoid fragmentation: 50
 Done
> set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes –dropFragCpuThreshold 50
 Done
> show iptunnelparam
Tunnel Source IP: 0.0.0.0
Round Robin of Tunnel Source IP: Yes
Drop if Fragmentation Needed: Yes
CPU usage threshold to avoid fragmentation: 50
 Done
```

Note: To create a new MIP or SNIP address to use as the global source IP address, use the add ns ip command before you type the set iptunnelparam command.

Parameters for customizing IP tunnels globally

srcIP

The common source IP address for all tunnels. Must be a MIP or a SNIP address.

srcIPRoundRobin

Use a different source IP address for each new session through a particular IP tunnel, as determined by round robin selection of one of the SNIP addresses. This setting is ignored if a common global source IP address has been specified for all the IP tunnels. This setting does not apply to a tunnel for which a source IP address has been specified. Possible values: YES, NO. Default: NO.

dropFrag

Drop any packet that requires fragmentation. Possible values: YES, NO. Default: NO.

dropFragCpuThreshold

Threshold value, as a percentage of CPU usage, at which to drop packets that require fragmentation. Applies only if dropFrag is set to NO. Minimum value: 1. Maximum value: 100. Default: 0 (Not set).

To globally customize IP tunnels by using the configuration utility

1. In the navigation pane, expand **Network**.
2. In the details pane, in the **Settings** group, click **IP Tunnel Global Settings**.
3. In the **Configure IP Tunnel Global Parameters** dialog box, set the following parameters, which correspond to parameters described in "Parameters for customizing the IP tunnels globally" as shown:
   - **Source IP**—srcIP
   - **Round Robin of Source IP**—srcIPRoundRobin
   - **Drop Packet if Fragmentation is required**—dropFrag
   - **Don’t fragment and drop packet if CPU usage is >=** —dropFragCpuThreshold
4. Click **OK** and then click **Close.** A message appears in the status bar, stating that the IP Tunnel Global Parameters have been configured successfully.

Interfaces

Before you begin configuring interfaces, decide whether your configuration can use MAC-based forwarding mode, and either enable or disable this system setting accordingly. The number of interfaces in your configuration is different for the different models of the Citrix® NetScaler® appliance. In addition to configuring individual interfaces, you can logically group interfaces, using VLANs to restrict data flow within a set of interfaces, and you can aggregate links into channels. In a high availability setup, you can configure a virtual MAC (VMAC) address if necessary. If you use L2 mode, you might want to modify the aging of the bridge table.

When your configuration is complete, decide whether you should enable the system setting for path MTU discovery. NetScaler appliances can be deployed in active-active mode using VRRP. An active-active deployment, in addition to preventing downtime, makes efficient use of all the NetScaler appliances in the deployment. You can use the Network Visualizer tool to view the network configuration of a NetScaler deployment and configure interfaces, channels, VLANs, and bridge groups.

Configuring MAC-Based Forwarding

With MAC-based forwarding (MBF) enabled, when a request reaches the NetScaler appliance, the appliance remembers the source MAC address of the frame and uses it as the destination MAC address for the resulting replies. MAC-based forwarding can be used to avoid multiple-route/ARP lookups and to avoid asymmetrical packet flows. MAC-based forwarding may be required when the NetScaler is connected to multiple stateful devices, such as VPNs or firewalls, because it ensures that the return traffic is sent to the same device that the initial traffic came from.

MAC-based forwarding is useful when you use VPN devices, because it guarantees that all traffic flowing through a VPN passes back through the same VPN device.

The following topology diagram illustrates the process of MAC-based forwarding.

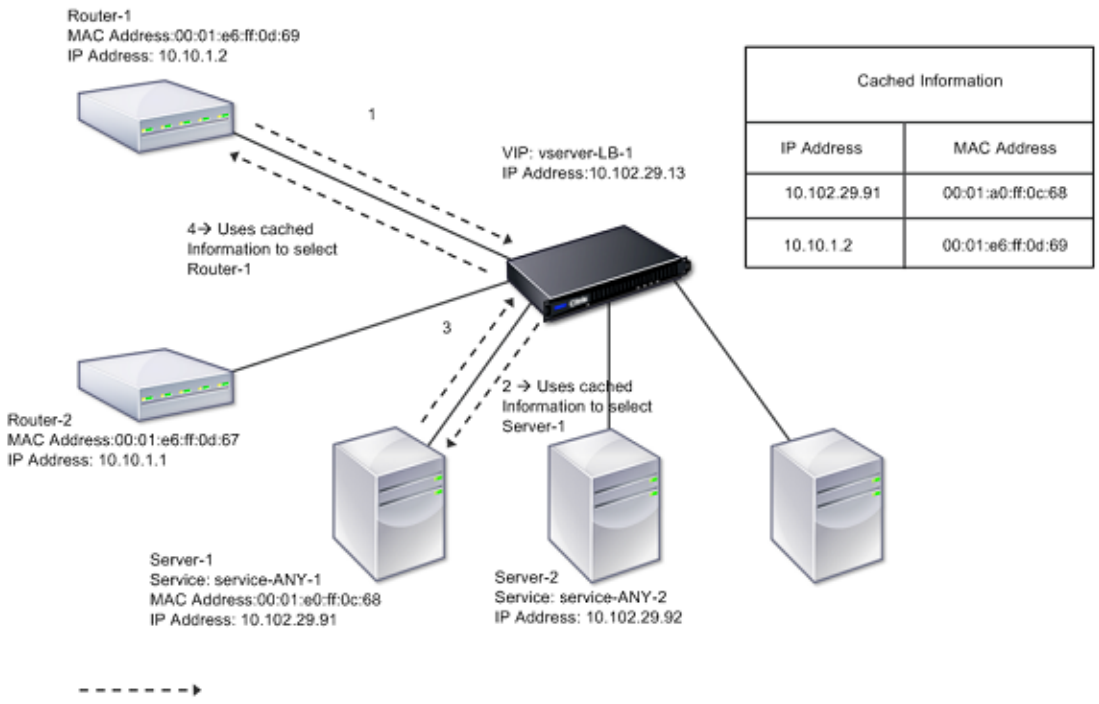


Figure 1. MAC-Based Forwarding Mode

When MAC-based forwarding (MBF) is enabled, the NetScaler caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server replies through the NetScaler appliance, the appliance sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows

in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when the NetScaler initiates a connection, it uses the route and ARP tables for the lookup function. In a direct server return configuration, you must enable MAC-based forwarding.

For more information about direct server return configurations, see "Load Balancing" chapter of the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Some deployment topologies may require the incoming and outgoing paths to flow through different routers. MAC-based forwarding would break this topology design.

MBF should be disabled in the following situations:

- **When you configure link load balancing.** In this case, asymmetric traffic flows are desirable because of link costs.
- **When a server uses network interface card (NIC) teaming without using LACP (802.1ad Link Aggregation).** To enable MAC-based forwarding in this situation, you must use a layer 3 device between the NetScaler and server.

  Note: MBF can be enabled when the server uses NIC teaming with LACP, because the virtual interface uses one MAC address.
- When firewall clustering is used. Firewall clustering assumes that ARP is used to resolve the MAC address for inbound traffic. Sometimes the inbound MAC address can be a non-clustered MAC address and should not be used for inbound packet processing.

When MBF is disabled, the NetScaler uses L2 or L3 connectivity to forward the responses from servers to the clients. Depending on the route table, the routers used for outgoing connection and incoming connection can be different. In the case of reverse traffic (response from the server):

- If the source and destination are on different IP subnets, the NetScaler uses the route lookup to locate the destination.
- If the source is on the same subnet as the destination, the NetScaler looks up the ARP table to locate the network interface and forwards the traffic to it. If the ARP table does not exist, the NetScaler requests the ARP entries.

To enable or disable MAC-based forwarding by using the NetScaler command line

At the NetScaler command prompt, type:

- enable ns mode mbf
- disable ns mode mbf

To enable or disable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, in the **Modes and Features** group, click **Configure modes**.
3. In the **Configure Modes** dialog box, do one of the following:
   - To enable MAC-based forwarding, select the **MAC-based forwarding** check box.
   - To disable MAC-based forwarding, clear the **MAC-based forwarding** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**. A message appears in the status bar, stating that the selected modes are enabled and the unselected modes are disabled.

Configuring Network Interfaces

Network interfaces in the NetScaler appliance are numbered in <slot>/<port> notation. After configuring your interfaces, you should display the interfaces and their settings to verify the configuration. You can also display this information to troubleshoot a problem in the configuration.

To manage the network interfaces, you might have to enable some interfaces and disable others. You can reset an interface to renegotiate its settings. You can clear the accumulated statistics for an interface. To verify the configuration, you can display the interface settings. You can display the statistics for an interface to evaluate its health.

Setting the Network Interface Parameters

The network interface configuration is neither synchronized nor propagated. For an HA pair, you must perform the configuration on each unit independently.

Network interface parameters include Link Aggregate Control Protocol (LACP) settings. For more information about Link Aggregate Control Protocol (LACP), see Configuring Link Aggregation Using the Link Aggregate Channel Protocol.

To set the network interface parameters by using the NetScaler command line

At the NetScaler command prompt, type:

- set interface <id> [**-speed** <speed>] [**-duplex** <duplex>] [**-flowControl** <flowControl>] [**-autoneg** (DISABLED | ENABLED)] [**-haMonitor** (ON | OFF)] [**-trunk** (ON | OFF)] [**-lacpMode** <lacpMode>] [**-lacpKey**<positive_integer>] [**-lacpPriority** <positive_integer>] [**-lacpTimeout** (LONG | SHORT)] [**-ifAlias** <string>] [**-throughput** <positive_integer>][**-bandwidthHigh** <positive_integer> [**-bandwidthNormal** <positive_integer>]]

- show interface [<id>]

Example

```
> set interface 1/8 -duplex full
 Done
> show interface 1/8
      Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
      flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg, 802.1q>
      MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h53m53s
      Requested: media UTP, speed AUTO, duplex FULL, fctl OFF, throughput 0
      RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
      TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
      NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
      Bandwidth thresholds are not set.
Done
```

Parameters for setting a Network Interface

id

The number assigned to the interface.

speed

Ethernet speed for the interface. Possible values: AUTO, 10, 100, 1000, and 10000 Mbps. Default: AUTO. A setting other than AUTO requires the same configuration for device at the other end of the link. Mismatched speed or duplex configurations can cause link errors, packet losses, and other errors. Some network interfaces do not support certain speeds. An attempt to set an unsupported speed is reported as an error.

duplex

Duplex mode for the interface. Possible values: AUTO, HALF, and FULL. Default: AUTO. AUTO is recommended. If you force HALF or FULL mode, you must manually configure the same mode and identical speed on both sides of the link.

flowControl

Apply 802.3x flow control to the interface. Possible values: OFF, RX, TX, RXTX, and ON (forced RXTX). Default: OFF. Real flow control status depends on the auto-negotiation results. Link parameter mismatches must be checked for and avoided because, for example, they can cause the NetScaler to drop packets, or the link may not be accessible.

autoneg

Use auto negotiation on the interface. Possible values: DISABLED and ENABLED.

haMonitor

Monitor the interface for failure events. Possible values: ON and OFF. Default: ON. When ON in an HA configuration, failover occurs when a network interface fails. If a network interface is not being used, or if failover is not required, select OFF. (Also, if the network interface is not used in the configuration, you must disable it.)

trunk

Trunk port functionality for the interface. Possible values: ON and OFF. Default: OFF. With the ON setting, traffic is tagged for the VLANs bound to this network interface, including the default VLAN. If you require 802.1q behavior with backward compatibility, you must set this parameter to OFF.

lacpMode

LACP mode. Possible values: DISABLED, ACTIVE, and PASSIVE. Default: DISABLED

lacpKey

LACP key for the interface. Possible values: 1 to 4.

lacpPriority

LACP port priority. Possible values: 1 to 65535. Default: 32768.

lacpTimeout

LACP timeout setting. Possible values: LONG and SHORT. Default: LONG.

ifAlias

Alias name for the interface.

throughput

Minimum required throughput for the interface.

To set the network interface parameters by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Interfaces**.
2. On the **Interfaces** pane, select the network interface that you want to modify (for example, **1/8**), and then click **Open**.
3. In the **Configure Interface** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a network interface" as shown:
   - **Speed**—speed
   - **Duplex**—duplex
   - **Flow Control**—flowControl
   - **Auto Negotiation**—autoneg
   - **HA Monitoring**—haMonitor
   - **Trunk**—trunk
   - **Alias Name**—ifAlias
   - **Throughput**—throughput
   - **Bandwidth High**—bandwidthHigh
   - **Bandwidth Normal**—bandwidthNormal
   - **LACP Mode**—lacpMode
   - **LACP Key**—lacpKey
   - **LACP Time**—out-lacpTimeout
   - **LACP Priority**—lacpPriority
4. Click **OK**. A message appears in the status bar, stating that the interface has been configured successfully.

Enabling and Disabling Network Interfaces

By default, the network interfaces are enabled. You must disable any network interface that is not connected to the network, so that it cannot send or receive packets. Disabling a network interface that is connected to the network in a high availability setup can cause failover.

For more information about high availability, see High Availability.

To enable or disable a network interface by using the NetScaler command line

At the NetScaler command prompt, type one of the following pairs of commands to enable or disable an interface and verify the setting:

- enable interface <interface_num>
- show interface <interface_num>
- disable interface <interface_num>
- show interface <interface_num>

Example

```
> enable interface 1/8
 Done
> show interface 1/8
      Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
      flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg, 802.1q>
      MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
      Requested: media UTP, speed AUTO, duplex FULL, fctl OFF, throughput 0
      RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
      TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
      NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
      Bandwidth thresholds are not set.
Done
```

To enable or disable a network interface by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Interfaces**.
2. On the **Interfaces** pane, select the network interface that you want to enable or disable, and do one of the following:
   - To enable a network interface, click **Enable**.
   - To disable a network interface, click **Disable**.

   A message appears in the status bar, stating that the network interface has been enabled or disabled successfully.

Resetting Network Interfaces

Network interface settings control properties such as duplex and speed. To renegotiate the settings of a network interface, you must reset it.

To reset a network interface by using the NetScaler command line

At the NetScaler command prompt, type the following commands to reset an interface and verify the setting:

- reset interface <interface_num>
- show interface <interface_num>

Example

```
> reset interface 1/8
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4004000 <disabled, DOWN, BOUND to LA/1, down, autoneg, 802.1q>
        MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 907h04m59s
        Requested: media UTP, speed AUTO, duplex FULL, fctl OFF, throughput 0
        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
```

To reset a network interface by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Interfaces**.
2. On the **Interfaces** pane, select the network interface that you want to reset (for example, **1/8**).
3. Click **Reset Interface**. A message appears in the status bar, stating that the network interface has been reset successfully.

Monitoring a Network Interface

You can display network interface statistics to monitor parameters such as packets sent and packets received, throughput, Link Aggregate Control Protocol (LACP) data units, and errors, and use the information to check the health of the network interface. You can clear the statistics of a network interface to monitor its statistics from the time the statistics are cleared.

To display the statistics of the network interfaces by using the NetScaler command line

At the NetScaler command prompt, type:

stat interface <interface_num>

Example

```
> stat interface 1/8
Interface [1/8]:
Interface State                 DOWN
Link uptime                  00:00:00
Link downtime                8.01:01:34

Throughput Statistics
                          Rate (/s)          Total
Bytes received                   0              0
Bytes transmitted                0              0
Packets received                 0              0
Packets transmitted              0              0

Packet Statistics
                          Rate (/s)          Total
Multicast packets                0              0
NetScaler packets                0              0

LACP Statistics
                          Rate (/s)          Total
LACPDUs received                 0              0
LACPDUs transmitted              0          23166

Error Statistics
                          Rate (/s)          Total
Error packets received (hw)      0              0
Error packets transmitted (hw)   0              0
```

```
Inbound packets discarded(hw)              0                0
Outbound packets discarded(hw)             0                0
Packets dropped in Rx (sw)                 0                0
Packets dropped in Tx (sw)                 0            23166
NIC hangs                                 --                0
Status stalls                             --                0
Transmit stalls                           --                0
Receive stalls                            --                0
Error-disables                            --                0
Duplex mismatches                         --                0
Link re-initializations                   --                3
MAC moves registered                       0                0
Times NIC become muted                    --                0
 Done
>
```

To display the statistics of an Interface by using the configuration utility

1. In the navigation pane, expand **Network** and click **Interfaces**.
2. On the **Interfaces** page, select the network interface whose statistics you want to display (for example, **1/8**).
3. Click **Statistics**.

To clear a network interface's statistics by using the NetScaler command line

At the NetScaler command prompt, type:

clear interface <interface_num>

Example

```
> clear interface 1/8
 Done
```

To clear a network interface's statistics by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Interfaces**.

2. On the **Interfaces** pane, select the network interface whose statistics youwant to clear (for example, **1/8**).

3. Click **Clear Statistics**. A message appears in the status bar, stating that the statistics have been successfully cleared.

Configuring Forwarding Session Rules

By default, the NetScaler appliance does not create session entries for traffic that it only forwards (L3 mode). For a case in which a client request that the appliance forwards to a server results in a response that has to return by the same path, you can create a forwarding-session rule. A forwarding-session rule creates forwarding-session entries for traffic that originates from or is destined for a particular network and is forwarded by the NetScaler.

Note: This feature is supported only on NetScaler 9.3.e.

To create a forwarding session rule by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a forwarding-session rule and verify the configuration:

- add forwardingSession <name> [<network> <netmask>] | [-**aclname** <string>]
- show forwardingSession

Examples

A network address as the condition:

```
> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done

> show forwardingSession fs-nw-1
1)      Forward Session: fs-nw-1
       Network: 10.102.105.51     Netmask: 255.255.255.255

 Done
```

An ACL as the condition:

```
> add forwardingSession fs-acl-1 acl1
 Done

> show forwardingSession fs-acl-1
1)      Forward Session: fs-acl-1
       ACL Name: acl1
 Done
```

Parameters for configuring a forwarding session rule

name (Name)

The name of the forwarding session rule that you are configuring. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

network (Subnet IP)

The network address from which the forwarded traffic originates or to which it is destined.

netmask (Netmask)

Subnet mask associated with the network.

aclname (ACL Name)

The name of an extended ACL with action set to ALLOW. The rule specified in the ACL is used as a forwarding-session rule.

To configure a forwarding session rule by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Forwarding Sessions**.
2. In the details pane, click **Add**.
3. In the **Create Forwarding Session** dialog box, set the **Name** parameter:
4. Do one of the following:
   - If you want to use the network address as a condition for creating a forwarding session rule, click **Subnet** and set the following parameters:
     - **Subnet IP**
     - **Netmask**
   - If you want to use an extended ACL as a condition for creating a forwarding session rule, click **ACL** and set the **ACL Name** parameter:
5. Click **Create**, and then **Close**. A message appears in the status bar, stating that the forwarding session rule has been configured successfully.

Understanding VLANs

A NetScaler appliance supports Layer 2 port and IEEE 802.1q tagged VLANs. VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface as a part of multiple VLANs by using IEEE 802.1q tagging.

You can configure VLANs and bind them to IP subnets. The NetScaler then performs IP forwarding between these VLANs (if it is configured as the default router for the hosts on these subnets).

The NetScaler supports the following types of VLANs:

Port-Based VLANs. The membership of a port-based VLAN is defined by a set of network interfaces that share a common, exclusive Layer 2 broadcast domain. You can configure multiple port-based VLANs. By default, all network interfaces on the NetScaler are members of VLAN 1.

If you apply 802.1q tagging to the port, the network interface belongs to a port-based VLAN. Layer 2 traffic is bridged within a port-based VLAN, and Layer 2 broadcasts are sent to all members of the VLAN if Layer 2 mode is enabled. When you add an untagged network interface as a member of a new VLAN, it is removed from its current VLAN.

Default VLAN. By default, the network interfaces on the NetScaler are included in a single, port-based VLAN as untagged network interfaces. This VLAN is the default VLAN. It has a VLAN ID (VID) of 1. This VLAN exists permanently. It cannot be deleted, and its VID cannot be changed.

When you add a network interface to a to a different VLAN as an untagged member, the network interface is automatically removed from the default VLAN. If you unbind a network interface from its current port-based VLAN, it is added to the default VLAN again.

Tagged VLANs. 802.1q tagging (defined in the IEEE 802.1q standard) allows a networking device (such as the NetScaler) to add information to a frame at Layer 2 to identify the VLAN membership of the frame. Tagging allows network environments to have VLANs that span multiple devices. A device that receives the packet reads the tag and recognizes the VLAN to which the frame belongs. Some network devices do not support receiving both tagged and untagged packets on the same network interface—in particular, Force10 switches. In such cases, you need to contact customer support for assistance.

The network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of one VLAN only (its native VLAN). This network interface transmits the frames for the native VLAN as untagged frames. A network interface can be a part of more than one VLAN if the other VLANs are tagged.

When you configure tagging, be sure to match the configuration of the VLAN on both ends of the link. The port to which the NetScaler connects must be on the same VLAN as the NetScaler network interface.

Note: This VLAN configuration is neither synchronized nor propagated, therefore you must perform the configuration on each unit in an HA pair independently.

Applying Rules to Classify Frames

VLANs have two types of rules for classifying frames:

Ingress rules. Ingress rules classify each frame as belonging only to a single VLAN. When a frame is received on a network interface, the following rules are applied to classify the frame:

- If the frame is untagged, or has a tag value equal to 0, the VID of the frame is set to the port VID (PVID) of the receiving interface, which is classified as belonging to the native VLAN. (PVIDs are defined in the IEEE 802.1q standard.)
- If frame has a tag value equal to FFF, the frame is dropped.
- If the VID of the frame specifies a VLAN of which the receiving network interface is not a member, the frame is dropped. For example, if a packet is sent from a subnet associated with VLAN ID 12 to a subnet associated with VLAN ID 10, the packet is dropped. If an untagged packet with VID 9 is sent from the subnet associated with VLAN ID 10 to a network interface PVID 9, the packet is dropped.

Egress Rules. The following egress rules are applied:

- If the VID of the frame specifies a VLAN of which the transmission network interface is not a member, the frame is discarded.
- During the learning process (defined by the IEEE 802.1q standard), the Src MAC and VID are used to update the bridge lookup table of the NetScaler.
- A frame is discarded if its VID specifies a VLAN that does not have any members. (You define members by binding network interfaces to a VLAN.)

VLANs and Packet Forwarding on the NetScaler

The forwarding process on the NetScaler appliance is similar to that on any standard switch. However, the NetScaler performs forwarding only when Layer 2 mode is on. The key features of the forwarding process are:

- Topology restrictions are enforced. Enforcement involves selecting each network interface in the VLAN as a transmission port (depending on the state of the network interface), bridging restrictions (do not forward on the receiving network interface), MTU restrictions, and so on.
- Frames are filtered on the basis of information in the bridge table lookup in the forwarding database (FDB) table of the NetScaler. The bridge table lookup is based on the destination MAC and the VID. Packets addressed to the MAC address of the NetScaler are processed at the upper layers.
- All broadcast and multicast frames are forwarded to each network interface that is a member of the VLAN, but forwarding occurs only if L2 mode is enabled. If L2 mode is disabled, the broadcast and multicast packets are dropped. This is also true for MAC addresses that are not currently in the bridging table.
- A VLAN entry has a list of member network interfaces that are part of its untagged member set. When forwarding frames to these network interfaces, a tag is not inserted

in the frame.

- If the network interface is a tagged member of this VLAN, the tag is inserted in the frame when the frame is forwarded.

When a user sends any broadcast or multicast packets without the VLAN being identified, that is, during duplicate address detection (DAD) for NSIP or ND6 for the next hop of the route, the packet is sent out on all the network interfaces, with appropriate tagging based on either the Ingress and Egress rules. ND6 usually identifies a VLAN, and a data packet is sent on this VLAN only. Port-based VLANs are common to IPv4 and IPv6. For IPv6, the NetScaler supports prefix-based VLANs.

Configuring a VLAN

You can implement VLANs in the following environments:

- Single subnet
- Multiple subnets
- Single LAN
- VLANs (no tagging)
- VLANs (802.1q tagging)

If you configure VLANs that have only untagged network interfaces as their members, the total number of possible VLANs is limited to the number of network interfaces available in the NetScaler. If more IP subnets are required with a VLAN configuration, 802.1q tagging must be used.

When you bind a network interface to a VLAN, the network interface is removed from the default VLAN. If the network interfaces need to be a part of more than one VLAN, you can bind the network interfaces to the VLANs as tagged members.

You can configure the NetScaler to forward traffic between VLANs at Layer 3. In this case, a VLAN is associated with a single IP subnet. The hosts in a VLAN that belong to a single subnet use the same subnet mask and one or more default gateways connected to that subnet. Configuring Layer 3 for a VLAN is optional. Layer 3 is used for IP forwarding (inter-VLAN routing). Each VLAN has a unique IP address and subnet mask that define an IP subnet for the VLAN. In an HA configuration, this IP address is shared with the other NetScaler appliances. The NetScaler forwards packets between configured IP subnets (VLANs).

When you configure the NetScaler, you must not create overlapping IP subnets. Doing so impedes Layer 3 functionality.

Each VLAN is a unique Layer 2 broadcast domain. Two VLANs, each bound to separate IP subnets, cannot be combined into a single broadcast domain. Forwarding traffic between two VLANs requires a Layer 3 forwarding (routing) device, such as the NetScaler appliance.

Creating or Modifying a VLAN

To configure a VLAN, you create a VLAN entity, and then bind network interfaces and IP addresses to the VLAN. If you remove a VLAN, its member interfaces are added to the default VLAN.

To create a VLAN by using the NetScaler command line

At the NetScaler command prompt, type:

add vlan <id> [**-aliasName** <string>] [**-ipv6DynamicRouting** (ENABLED|DISABLED)]

Example

add vlan 2 –aliasName “Network A”

To bind an interface to a VLAN by using the NetScaler command line

At the NetScaler command prompt, type:

bind vlan <id> **-ifnum** <slot/port>

Example

bind vlan 2 -ifnum 1/8

To bind an IP address to a VLAN by using the NetScaler command line

At the NetScaler command prompt, type:

bind vlan <id> **-IPAddress** <IPAddress> <netMask>

Example

bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0

To remove a VLAN by using the NetScaler command line

At the NetScaler command prompt, type:

rm vlan <id>

Parameters for configuring a VLAN

id

An integer that uniquely identifies the VLAN to which a particular frame belongs. The NetScaler supports a maximum of 4094 VLANs. ID 1 is reserved for the default VLAN. Minimum value: 2. Maximum value: 4094.

aliasName

A name for the VLAN. Must begin with a letter, a number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the VLAN. However, you cannot perform any VLAN operation by specifying the alias name instead of the VLAN ID.

ipv6DynamicRouting

Enable or disable IPv6 dynamic routing on this VLAN. Possible values: ENABLED, DISABLED. Default: DISABLED.

ifNum

The name, in <slot>/<port> notation, of an interface to be bound to the VLAN.

IPAddress

The IP address that is to be assigned to the VLAN. An entry for the subnet must be in the routing table before you issue this command. Overlapping subnets are not allowed. The IP address specified can be used as the default gateway among the hosts in the subnet to

allow for IP forwarding between VLANs.

Caution: DO NOT specify an IP address for VLAN 1.

netMask

Defines the network mask for the subnet defined for this VLAN.

To configure a VLAN by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **VLANs**.
2. In the details pane, do one of the following:
   - To create a new VLAN, click **Add**.
   - To modify an existing VLAN, click **Open**.
3. In the **Add VLAN** or **Configure/Modify VLAN** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a VLAN” as shown:
   - **VLAN ID***—id
   - **Enable IPv6 dynamic routing**—ipv6DynamicRouting
   - **Alias Name**—aliasName

   * A required parameter
4. To bind an IP address to a VLAN, under **IPs**, select the **Active** check box corresponding to the IP address that you want to bind to the VLAN (for example, **10.102.29.54**). The **Type** column displays the IP address type (such as mapped IP, virtual IP, or subnet IP) for each IP address in the **IP Addresses** column.
5. To bind a network interface to a VLAN, under **Interfaces**, select the **Active** check box corresponding to the interface that you want to bind to the VLAN (for example, **1/8**).
6. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the VLAN has been configured successfully.

Monitoring VLANS

You can display VLAN statistics such as packets received, bytes received, packets sent, and bytes sent, and use the information to identify anomalies and or debug a VLAN.

To view the statistics of a VLAN by using the NetScaler command line

At the NetScaler command prompt, type:

stat vlan <vlanID>

Example

stat vlan 2

To view the statistics of a VLAN by using the configuration utility

1. In the navigation pane, expand **Network** and click **VLANs**.
2. On the **VLANs** page, select the VLAN whose statistics you want to view (for example, **2**).
3. Click **Statistics**.

Configuring VLANs in an HA Setup

VLAN configuration for a high-availability setup requires that the NetScaler appliances have the same hardware configuration, and the VLANs configured on them must be mirror images.

The correct VLAN configuration is implemented automatically when the configuration is synchronized between the NetScaler appliances. The result is identical actions on all the appliances. For example, adding network interface 0/1 to VLAN2 adds this network interface to VLAN 2 on all the appliances participating in the high-availability setup.

Note: If you use network-interface-specific commands in an HA setup, the configurations you create are not propagated to the other NetScaler appliance. You must perform these commands on each appliance in an HA pair to ensure that the configuration of the two appliances in the HA pair remains synchronized.

Configuring VLANs on a Single Subnet

Before configuring a VLAN on a single subnet, make sure that Layer 2 Mode is enabled.

The following figure shows a single subnet environment

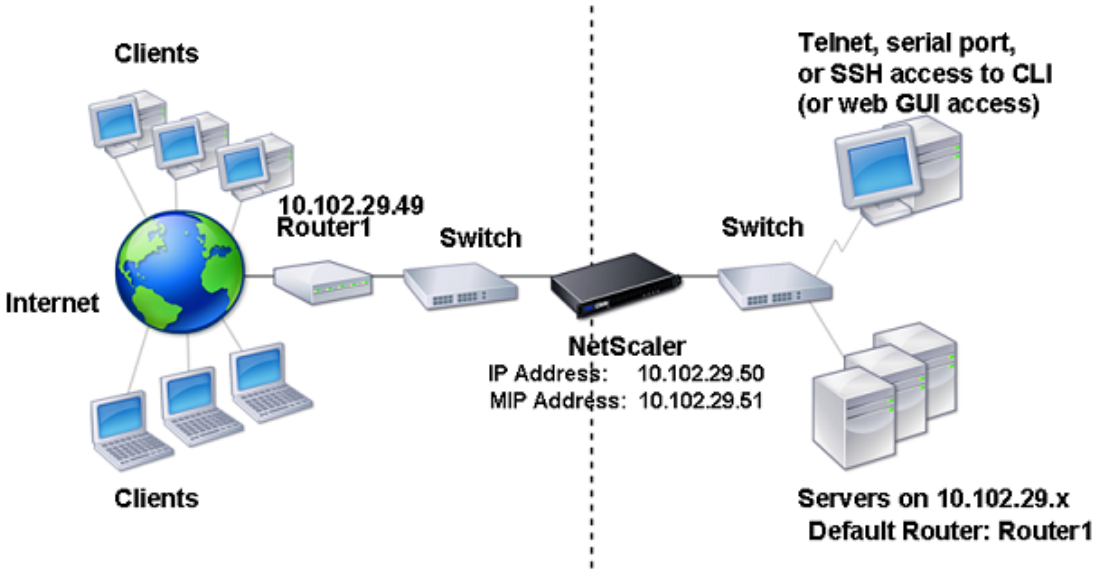


Figure 1. VLAN on a Single Subnet

In the above figure:

1. The default router for the NetScaler and the servers is Router 1.

   Layer 2 mode must be enabled on the NetScaler for the NetScaler to have direct access to the servers.

3. For this subnet, a virtual server can be configured for load balancing on the NetScaler.

To configure a VLAN on a single subnet, follow the procedures described in Creating or Modifying a VLAN. VLAN configuration parameters are not required, because the network interfaces are members of this VLAN.

Configuring VLANs on Multiple Subnets

To configure a single VLAN across multiple subnets, you must add a VIP for the VLAN and configure the routing appropriately. The following figure shows a single VLAN configured across multiple subnets.

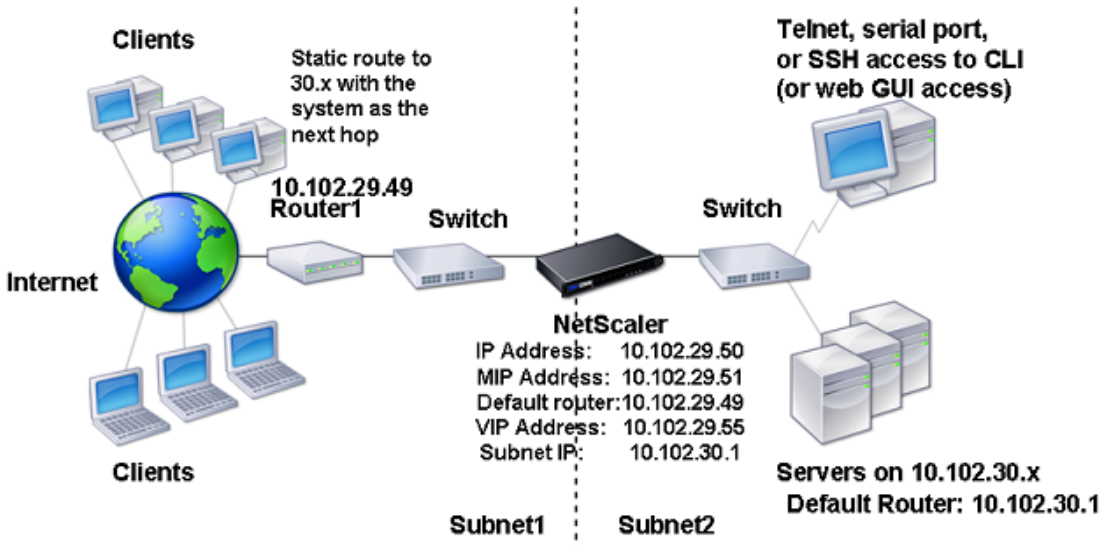


Figure 1. Multiple Subnets in a Single VLAN

To configure a single VLAN across multiple subnets, perform the following tasks:

Disable Layer 2 mode. For the procedure to disable Layer 2 mode, see the "Configuring System Management Settings" chapter of the *Citrix NetScaler Getting Started Guide* at http://support.citrix.com/article/CTX128672.

Add a VIP.

For the procedure to add a VIP, see Configuring and Managing Virtual IP Addresses (VIPs).

Configure RNAT ID.

For the procedure to configure the RNAT ID, see Configuring RNAT.

Configuring Multiple Untagged VLANS across Multiple Subnets

In environments with multiple untagged VLANs across multiple subnets, a VLAN is configured for each IP subnet. A network interface is bound to one VLAN only. The following figure shows this configuration.

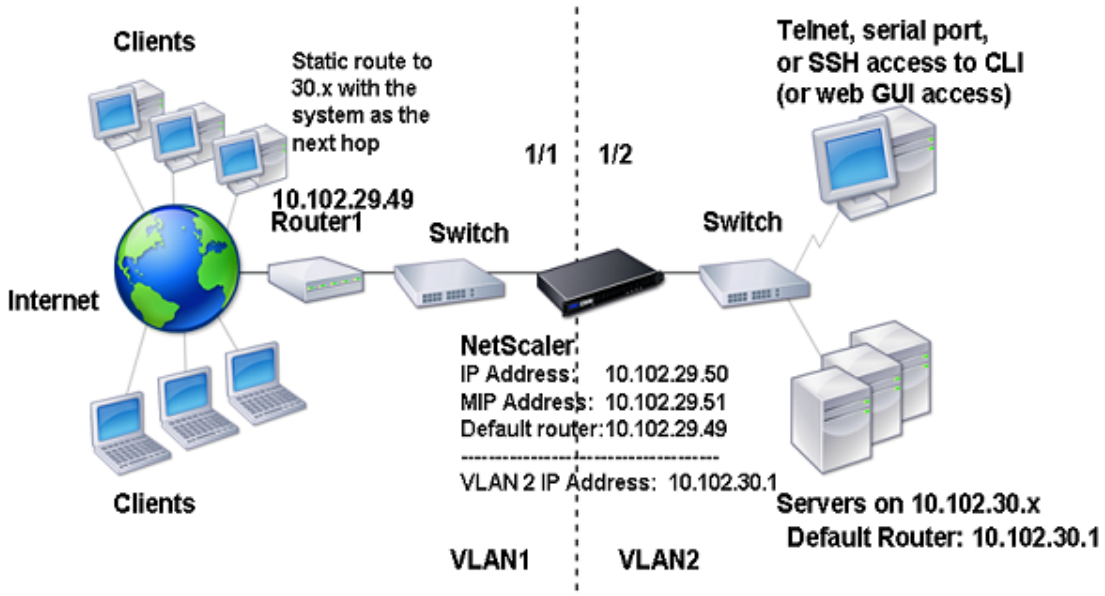


Figure 1. Multiple Subnets with VLANs - No Tagging

To implement the configuration shown in the above figure, perform the following tasks:

Add VLAN 2.

For the procedure to create a VLAN, see Creating or Modifying a VLAN.

Bind the 1/2 network interface of the NetScaler to VLAN 2 as an untagged network interface.

For the procedure to bind a network interface to a VLAN, see Creating or Modifying a VLAN.

Bind the IP address and subnet mask to VLAN 2.

For the procedure to bind a network interface to a VLAN, see Creating or Modifying a VLAN.

Configuring Multiple VLANs with 802.1q Tagging

For multiple VLANs with 802.1q tagging, each VLAN is configured with a different IP subnet. Each network interface is in one VLAN. One of the VLANs is set up as tagged. The following figure shows this configuration.

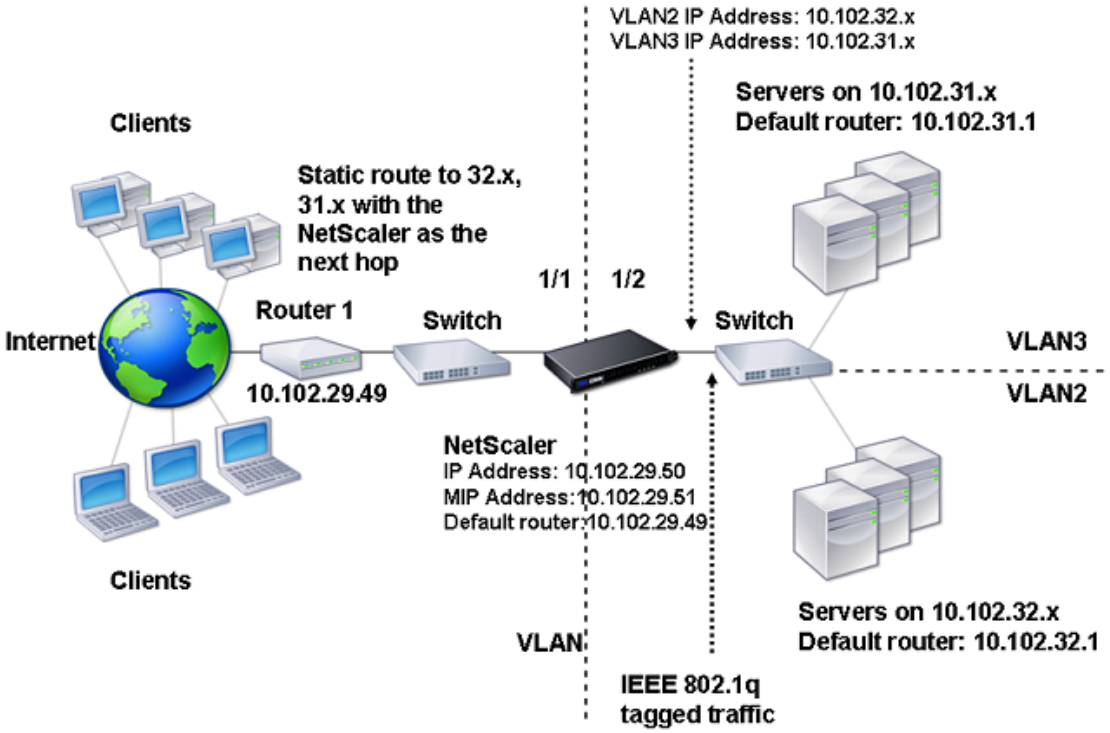


Figure 1. Multiple VLANs with IEEE 802.1q Tagging

To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.

   For the procedure to create a VLAN, see Creating or Modifying a VLAN.

2. Bind the 1/2 network interface of the NetScaler to VLAN 2 as an untagged network interface.

   For the procedure to bind a network interface to a VLAN, see Creating or Modifying a VLAN.

3.

Bind the IP address and netmask to VLAN 2.

For the procedure to bind an IP address to a VLAN, see Creating or Modifying a VLAN.

4. Add VLAN 3.

   For the procedure to create a VLAN, see Creating or Modifying a VLAN.

5. Bind the 1/2 network interface of the NetScaler to VLAN 3 as a tagged network interface.

   For the procedure to bind a network interface to a VLAN, see Creating or Modifying a VLAN.

   For the procedure to bind a tagged network interface, see Creating or Modifying a VLAN.

6. Bind the IP address and netmask to VLAN 3.

   For the procedure to bind an IP address to a VLAN, see Creating or Modifying a VLAN.

Configuring NSVLAN

NSVLAN is a VLAN to which the NetScaler management IP (NSIP) address's subnet is bound. The NSIP subnet is available only on interfaces that are associated with NSVLAN. By default, NSVLAN is VLAN1, but you can designate a different VLAN as NSVLAN. If you do so, you must reboot the NetScaler appliance for the change to take effect. After the reboot, NSIP subnet traffic is restricted to the new NSVLAN.

The traffic from the NetScaler IP subnet can be tagged (802.1q) with the VLAN ID specified for NSVLAN. You must configure the attached switch interface to tag and allow this same VLAN ID on the connected interface.

If you remove your NSVLAN configuration, the NSIP subnet is automatically bound to VLAN1, restoring the default NSVLAN.

To configure NSVLAN by using the NetScaler command line

At the NetScaler command prompt, type:

- set ns config -**nsvlan** <positive_integer> -**ifnum** <interface_name> ... [-**tagged** (YES|NO)]
- show ns config

Note: The configuration will take effect after the NetScaler appliance is rebooted.

Example

```
> set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged NO
 Done

> save config
 Done

> show ns config
      NetScaler IP: 10.102.29.170  (mask: 255.255.255.0)
      Number of MappedIP(s): 6
      Node: Standalone
      NetScaler IP Vlan: 300 Tagged: NO Bound Ports: 1/1 ½ 1/3

      Global configuration settings:
                  HTTP port(s): (none)
              Max connections: 0
        Max requests per connection: 0
```

```
                Client IP insertion: DISABLED
                    Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                      Min Path MTU: 576
              Path MTU entry timeout: 10
                    FTP Port Range: 0
                     CR Port Range: 0
                         Timezone: GMT+05:30-IST-Asia/Colombo
                       System Time: Tue Feb 22 16:50:44 2011
            Last Config Changed Time: Tue Feb 22 16:48:02 2011
              Last Config Saved Time: Tue Feb 22 16:48:19 2011
WARNING: The configuration must be saved and the system rebooted for these settings to take effect
 Done
```

To restore the default NSVLAN configuration by using the NetScaler command line

At the NetScaler command prompt, type:

- unset ns config -**nsvlan**
- show ns config

Example

```
> unset ns config -nsvlan
 Done

> sh ns config
      NetScaler IP: 10.102.29.170  (mask: 255.255.255.0)
      Number of MappedIP(s): 6
      Node: Standalone

      Global configuration settings:
                    HTTP port(s): (none)
                  Max connections: 0
        Max requests per connection: 0
              Client IP insertion: DISABLED
                  Cookie version: 0
      Persistence Cookie Secure Flag: ENABLED
                    Min Path MTU: 576
            Path MTU entry timeout: 10
                  FTP Port Range: 0
                   CR Port Range: 0
                       Timezone: GMT+05:30-IST-Asia/Colombo
                     System Time: Mon Feb 28 11:04:48 2011
          Last Config Changed Time: Mon Feb 28 11:04:40 2011
            Last Config Saved Time: Mon Feb 28 10:14:30 2011
 Done
```

Parameters for configuring NSVLAN

nsvlan

An integer that uniquely identifies the NSVLAN. Minimum value: 2. Maximum value: 4094.

ifNum

The name, in <slot>/<port> notation, of an interface to be bound to NSVLAN.

tagged

Designate all interfaces associated with NSVLAN as 802.1q tagged interfaces. The appliance adds a four-byte 802.1q tag to every packet sent on one of these interfaces. The tag identifies the VLAN. Possible values: YES, NO. Default: YES.

To configure NSVLAN by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change NSVLAN Settings**.
3. In the **Configure NSVLAN Settings** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring NSVLAN" as shown:
   - **NSVLAN ID**—nsvlan
   - **Tagged**—tagged

   *A required parameter
4. Under **Interfaces**, select interfaces from the **Available Interfaces** list and click **Add** to move them to the **Configured Interfaces** list.
5. Click **OK**. In the **Warning** dialog box, click **OK**. The configuration takes effect after the NetScaler appliance is rebooted.

Configuring Bridge Groups

Typically, when you want to merge two or more VLANs into a single domain, you change the VLAN configuration on all the devices in the separate domains. This can be a tedious task. To more easily merge multiple VLANs into a single broadcast domain, you can use bridge groups.

The bridge groups feature works the same way as a VLAN. Multiple VLANS can be bound to a single bridge group, and all VLANs bound to same bridge group form a single broadcast domain. You can bind only Layer 2 VLANs to a bridge group. For Layer 3 functionality, you must assign an IP address to a bridge group.

In Layer 2 mode, a broadcast packet received on an interface belonging to a particular VLAN is bridged to other VLANs that belong to the same bridge group. In the case of a unicast packet, the NetScaler appliance searches its bridge table for the learned MAC addresses of all the VLANs belonging to same bridge group.

In Layer 3 forwarding mode, an IP subnet is bound to a bridge group. The NetScaler accepts incoming packets belonging to the bound subnet and forwards the packets only on VLANs that are bound to the bridge group.

IPv6 routing can be enabled on a configured bridge group.

To add a bridge group and bind VLANs by using the NetScaler command line

To add a bridge group and bind VLANs and verify the configuration, type the following commands:

- add bridgegroup <id> [**-ipv6DynamicRouting** (ENABLED | DISABLED)]
- show bridgegroup <id>
- bind bridgegroup <id> **-vlan** <positive_integer>
- show bridgegroup <id>

Example

```
> add bridgegroup 12
 Done
> show bridgegroup 12
1)    Bridge Group: 12
      Member Interfaces : None
```

```
      Member vlans:
 Done
> bind bridgegroup 12 -vlan 4
 Done
> show bridgegroup 12
1)     Bridge Group: 12
       Member Interfaces : 1/8        Tagged: None
       Member vlans: 4
 Done
```

To remove a bridge group by using the NetScaler command line

At the NetScaler command prompt, type:

rm bridgegroup <id>

Example

```
 rm bridgegroup 12
```

Parameters for configuring bridge groups

id

A unique number that identifies a bridge group. Possible values: 1 to 1000.

vlan

The ID of a VLAN to be bound to the bridge group.

-ipv6DynamicRouting

Enable or disable IPv6 dynamic routing on this bridge group. Possible values: ENABLED, DISABLED. Default: DISABLED

To configure a bridge group by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Bridge Groups**.
2. In the details pane, do one of the following:
   - To create a new bridge group, click **Add**.
   - To modify an existing bridge group, click **Open**.
3. In the **Create Bridge Group** or **Configure Bridge Group** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring bridge groups" as shown:
   - **Bridge Group Id***—id
   - **Enable IPv6 dynamic routing**—ipv6DynamicRouting

   * A required parameter
4. To bind a VLAN to a bridge group, under **VLANs**, select the **Active** check box corresponding to the interface that you want to bind to the bridge group (for example, **1/8**).
5. To bind an IP address to a bridge group, under **IPs**, select the **Active** check box corresponding to the IP address that you want to bind to the bridge group (for example, **10.102.29.54**).
6. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the bridge group has been configured successfully.

Configuring VMACs

The primary and secondary nodes in a high availability (HA) setup share the Virtual MAC address (VMAC) floating entity. The primary node owns the floating IP addresses (such as MIP, SNIP, and VIP) and responds to ARP requests for these IP addresses with its own MAC address. Therefore, the ARP table of an external device, such as an upstream router, is updated with the floating IP address and the MAC address of the primary node.

When a failover occurs, the secondary node takes over as the new primary node. The former secondary node uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it had learned from the old primary node. The MAC address that the new primary node advertises is the MAC address of its own network interface. Some devices (a few routers) do not accept these GARP messages. Therefore, these external devices retain the IP address-to-MAC address mapping that the old primary node had advertised. This can result in a GSLB site going down.

Therefore, you must configure a VMAC on both nodes of an HA pair. This means that both nodes have identical MAC addresses. When a failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

For the procedures to configure a VMAC, see High Availability.

Configuring Link Aggregation

Link aggregation combines data coming from multiple ports into a single high-speed link. Configuring link aggregation increases the capacity and availability of the communication channel between the NetScaler appliance and other connected devices. An aggregated link is also referred to as a "channel." You can configure the channels manually, or you can use Link Aggregation Control Protocol (LACP). You cannot apply LACP to a manually configured channel, nor can you manually configure a channel created by LACP.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. (That is, the network interface parameters are ignored.) A network interface can be bound only to one channel.

When a network interface is bound to a channel, it drops its VLAN configuration. When network interfaces are bound to a channel, either manually or by LACP, they are removed from the VLANs that they originally belonged to and added to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you bind the network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then bind them to a channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them back to VLAN 2.

Configuring Link Aggregation Manually

When you create a link aggregation channel, its state is DOWN until you bind an active interface to it. You can modify a channel at any time. You can remove channels, or you can enable/disable them.

To create a link aggregation channel by using the NetScaler command line

At the NetScaler command prompt, type:

- add channel <id> [**-ifnum** <interfaceName> ...] [**-state** (ENABLED | DISABLED)] [**-speed** <speed>] [**-flowControl** <flowControl>] [**-haMonitor** (ON | OFF)][**-tagall** (ON | OFF)] [**-ifAlias** <string>] [**-throughput** <positive_integer>] [**-bandwidthHigh** <positive_integer> [**-bandwidthNormal** <positive_integer>]]
- show channels

Example

```
add channel LA/1 -ifnum 1/8
show channels
```

To bind an interface to or unbind an interface from an existing link aggregation channel by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- bind channel <id> <interfaceName>
- unbind channel <id> <interfaceName>

Example

bind channel LA/1 1/8

To modify a link aggregation channel by using the NetScaler command line

At the NetScaler command prompt, type the set channel command, the channel ID, and the parameters to be changed, with their new values.

Parameters for configuring a link aggregation channel

id

LA channel name, in form LA/*

(* An ID number for this channel)

ifnum

The name, in <slot>/<port> notation, of an interface to be bound to the channel.

state

Initial state for the channel. Possible values: ENABLED, DISABLED. Default: ENABLED.

speed

Speed for the channel. Possible values: AUTO, 10, 100, and 1000, and 10000. Default value:AUTO.

flowControl

Flow control for the channel. Possible values: OFF, RX, TX, and RXTX. Default value: OFF.

haMonitor

HA-monitoring control for the channel. Possible values: ON and OFF. Default value: ON.

tagall

Make this port a trunk port. When ON, port membership in all VLANs is tagged. If 802.1q behavior with native VLAN is required, use the OFF setting. Possible values: ON, OFF. Default: OFF.

ifAlias

Alias name for the channel. Maximum Length: 31.

throughput

Minimum required throughput for the network interface.

bandwidthHigh

Configured high threshold of the interface bandwidth usage in Mbps. An SNMP Trap message is generated if bandwidth usage of the interface crosses this limit. This parameter can be set only by using NetScaler command line.

bandwidthNormal

Configured normal threshold of the interface bandwidth usage in Mbits/s. A trap is generated if bandwidth usage of the interface returns to this level after exceeding the bandWidthHigh limit. This parameter can be set only by using NetScaler command line.

To configure a link aggregation channel by using the configuration utility

1. In the navigation pane, expand **Network** and click **Channels**.
2. In the details pane, do one of the following:
   - To create a new link aggregation channel, click **Add**.
   - To modify an existing link aggregation channel, select a channel and then click **Open**.
3. In the **Create Channel** or **Configure Channel** dialog box, specify values for the following parameters, which correspond to paremeters described in “Parameters for configuring a link aggregation channel as shown:
   - **Channel ID*****—id (Select a channel name from the drop-down list.)
   - **State**—state
   - **Throughput**— throughput
4. To bind an interface to the channel, on the **Bind/Unbind** tab, select an interface (for example, **1/8**) and click **Add**. (To remove an interface, select it and click **Remove**.)
5. Optionally, on the **Settings** tab, specify values for the following parameters, which correspond to paremeters described in “Parameters for configuring a link aggregation channel” as shown:
   - **Speed**—speed
   - **Flow Control**—flowControl
   - **HA Monitoring**—haMonitor
   - **Tag all VLANs**— tagall
   - **Alias Name**—ifAlias

To remove a link aggregation channel by using the NetScaler command line

Important: When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.

At the NetScaler command prompt, type:

rm channel <id>

Example

rm channel LA/1

To remove a link aggregation channel by using the configuration utility

Important: Important: When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.

1. In the navigation pane, expand **Network** and click **Channels**.
2. In the details pane, select the channel that you want to remove (for example, **LA/1**), and click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Configuring Link Aggregation by Using the Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) enables network devices to exchange link aggregation information by exchanging LACP Data Units (LACPDUs). Therefore, you cannot enable LACP on network interfaces that are members of a channel that you created manually.

When using LACP to configure link aggregation, you use different commands and parameters for modifying link aggregation channels than you do for creating link aggregation channels. To remove a channel, you must disable LACP on all interfaces that are part of the channel.

Note: In an High Availability configuration, LACP configurations are neither propagated nor synchronized.

Creating Link Aggregation Channels

For creating a link aggregation channel by using LACP, you need to enable LACP and specify the same LACP key on each interface that you want to be the part of the channel. For example, if you enable LACP and set the LACP Key to 3 on interfaces 1/1 and 1/2, a link aggregation channel LA/3 is created and interfaces 1/1 and 1/2 are automatically bound to it.

Note: When enabling LACP on a network interface, you must specify the LACP Key.

By default, LACP is disabled on all network interfaces.

To create an LACP channel by using the NetScaler command line

At the NetScaler command prompt, type:

- set interface <id> [**-lacpMode** <lacpMode>] [**-lacpKey**<positive_integer>] [**-lacpPriority** <positive_integer>] [**-lacpTimeout** (LONG | SHORT)]
- show interface [<id>]

Parameters for creating an LACP channel

id

The number assigned to the interface.

LacpMode

LACP mode. Possible values: DISABLED, ACTIVE, and PASSIVE. Default: DISABLED

lacpKey

LACP key for the interface. Possible values: 1 to 4.

lacpPriority

LACP port priority. Possible values: 1 to 65535. Default: 32768.

lacpTimeout

LACP timeout setting. Possible values: LONG and SHORT. Default: LONG.

To create an LACP channel by using the NetScaler the configuration utility

1. In the navigation pane, expand **Network**, and then click **Interfaces**.
2. On the **Interfaces** pane, select the network interface that you want to modify (for example, **1/8**), and then click **Open**.
3. In the **Configure interface** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a LACP channel" as shown:
   - **LACP Mode**—lacpMode
   - **LACP Key**—lacpKey
   - **LACP Time-out**—lacpTimeout
   - **LACP Priority**—lacpPriority
4. Click **OK**. A message appears in the status bar, stating that the interface has been configured successfully.

Modifying Link aggregation Channels

After you have created an LACP channel by specifying interfaces, you can modify properties of the channel.

To modify an LACP channel using the NetScaler command line.

At the NetScaler command prompt, type:

- set channel <id> [**-ifnum** <interfaceName> ...] [**-state** (ENABLED | DISABLED)] [**-speed** <speed>] [**-flowControl** <flowControl>] [**-haMonitor** (ON | OFF)] [**-ifAlias** <string>] [**-throughput** <positive_integer>] [**-tagall** (ON | OFF)] [**-bandwidthHigh** <positive_integer> [**-bandwidthNormal** <positive_integer>]]
- show channels

Example

set channel LA/3 –state ENABLED –speed 10000
show channels

Parameters for modifying an LACP channel

id

LA channel name, in form LA/*

(* An ID number for this channel)

state

Initial state for the channel. Possible values: ENABLED, DISABLED. Default: ENABLED.

speed

Speed for the channel. Possible values: AUTO, 10, 100, and 1000, and 10000. Default value:AUTO.

flowControl

Flow control for the channel. Possible values: OFF, RX, TX, and RXTX. Default value: OFF.

haMonitor

HA-monitoring control for the channel. Possible values: ON and OFF. Default value: ON.

tagall

Make this port a trunk port. When ON, port membership in all VLANs is tagged. If 802.1q behavior with native VLAN is required, use the OFF setting. Possible values: ON, OFF. Default: OFF.

ifAlias

Alias name for the channel. Maximum Length: 31.

throughput

Minimum required throughput for the network interface.

bandwidthHigh

Configured high threshold of the interface bandwidth usage in Mbps. An SNMP Trap message is generated if bandwidth usage of the interface crosses this limit. This parameter can be set only by using NetScaler command line.

bandwidthNormal

Configured normal threshold of the interface bandwidth usage in Mbits/s. A trap is generated if bandwidth usage of the interface returns to this level after exceeding the bandWidthHigh limit. This parameter can be set only by using NetScaler command line.

To modify an LACP channel by using the configuration utility

1. In the navigation pane, expand **Network** and click **Channels**.
2. In the details pane, select a LACP channel and then click **Open**.
3. In the **Configure Channel** dialog box, specify values for the following parameters, which correspond to paremeters described in “Parameters for configuring a link aggregation channel as shown:
   - **State**—state
   - **Throughput**— throughput
4. Optionally, on the **Settings** tab, specify values for the following parameters, which correspond to paremeters described in “Parameters for modifying an LACP channel” as shown:
   - **Speed**—speed
   - **Flow Control**—flowControl
   - **HA Monitoring**—haMonitor
   - **Tag all VLANs**— tagall
   - **Alias Name**—ifAlias

Removing a Link Aggregation Channel

To remove a link aggregation channel that was created by using LACP, you need to disable LACP on all the interfaces that are part of the channel.

To remove an LACP channel by using the NetScaler command line

At the NetScaler command prompt, type:

- set interface <id> **-lacpMode** Diasble
- show interface [<id>]

To remove an LACP channel by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **Interfaces**.
2. On the **Interfaces** pane, select the network interface that you want to modify (for example, **1/8**), and then click **Open**.
3. In the **Configure Interface** dialog box, clear the **Enable LACP** check box.
4. Click **OK**. A message appears in the status bar, stating that the interface has been configured successfully.

Binding an SNIP address to an Interface

You can now bind a NetScaler owned SNIP address to an interface without using Layer 3 VLANs. Any packets related to the SNIP address will go only through the bound interface.

Note: This feature is supported only on NetScaler 9.3.e.

This feature can be useful in a scenario where the upstream switch does not support Link Aggregation channels and you want the NetScaler appliance to load balance traffic, originated from a server, across the four links to the upstream switch as shown in the following illustration.

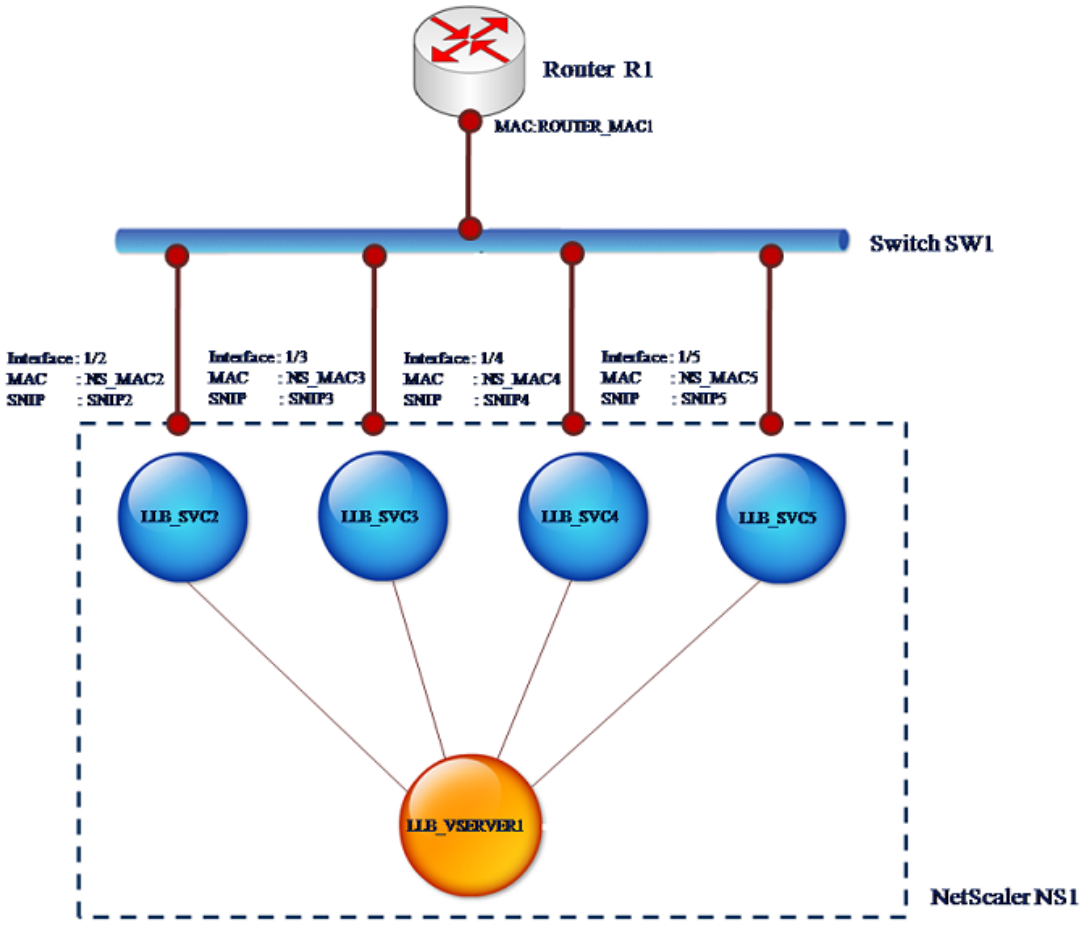


Figure 1.

The following tables describe the example settings for the scenario:

Entity	Name	Value

SNIP addresses on NS1	SNIP2 (for reference purpose only)	10.10.10.2
	SNIP3 (for reference purpose only)	10.10.10.3
	SNIP4 (for reference purpose only)	10.10.10.4
	SNIP5 (for reference purpose only)	10.10.10.5
LLB virtual server on NS1	LLB_VSERVER1	-
Transparent monitor on NS1	TRANS_MON	-
LLB services on NS1	LLB_SVC2	10.10.10.240
	LLB_SVC3	10.10.10.120
	LLB_SVC4	10.10.10.60
	LLB_SVC5	10.10.10.30
MAC address of interface 1/2 on NS1	NS_MAC_2 (for reference purpose only)	00:e0:ed:0f:bc:e0
MAC address of interface 1/3 on NS1	NS_MAC_3 (for reference purpose only)	00:e0:ed:0f:bc:df
MAC address of interface 1/4 on NS1	NS_MAC_4 (for reference purpose only)	00:e0:ed:0f:bc:de
MAC address of interface 1/5 on NS1	NS_MAC_5 (for reference purpose only)	00:e0:ed:1c:89:53
IP address of Router R1	Router_IP (for reference purpose only)	10.10.10.1
MAC address of interface of R1	ROUTER_MAC1 (for reference purpose only)	00:21:a1:2d:db:cc

To configure the example settings

1. Add four different SNIPs in different subnet ranges. This is for ARP to be resolved on four different links. For more information on creating a SNIP address, see Configuring Subnet IP Addresses (SNIPs).

   Command Line Interface example

   ```
   > add ns ip 10.10.10.2 255.255.255.0 -type SNIP
    Done
   > add ns ip 10.10.10.3 255.255.255.128 –type SNIP
   Done
   > add ns ip 10.10.10.4 255.255.255.192 –type SNIP
   Done
   > add ns ip 10.10.10.5 255.255.255.224 –type SNIP
   Done
   ```

2. Add four different dummy services in the added SNIP subnets. This is to ensure that the traffic is sent out with source IP as one of the four configured SNIPs. For more information on creating a service, see Configuring Services

   Command Line Interface example

   ```
   > add service LLB_SVC2 10.10.10.240 any *
    Done
   > add service LLB_SVC3 10.10.10.120 any *
    Done
   > add service LLB_SVC4 10.10.10.60 any *
    Done
   > add service LLB_SVC5 10.10.10.30 any *
    Done
   ```

3. Add a transparent ping monitor for monitoring the gateway. Bind the monitor to each of the configured dummy services. This is to make the state of the services as UP. For more information on creating a transparent monitor, see Creating and Binding a Transparent Monitor.

   Command Line Interface example

   ```
   > add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
    Done
   > bind monitor TRANS_MON LLB_SVC2
    Done
   > bind monitor TRANS_MON LLB_SVC3
    Done
   > bind monitor TRANS_MON LLB_SVC4
    Done
   > bind monitor TRANS_MON LLB_SVC5
    Done
   ```

4. Add a link load balancing (LLB) virtual server and bind the dummy services to it. For more information on creating an LLB virtual server, see Configuring an LLB Virtual Server and Binding a Service.

   Command Line Interface example

   ```
   > add lb vserver LLB_VSERVER1 any
    Done
   > set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   ```

5. Add the LLB virtual server as the default LLB route. For more information on creating an LLB route see Configuring an LLB Route.

   Command Line Interface example

   ```
   > add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
    Done
   ```

6. Add an ARP entry for each of the dummy services with the MAC address of the gateway. This way the gateway is reachable through these dummy services. For more information on adding an ARP entry, see Configuring Static ARP.

   Command Line Interface example

   ```
   > add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum 1/2
    Done
   > add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum 1/3
    Done
   > add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
    Done
   > add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
    Done
   ```

7. Bind a specific interface to an SNIP by adding an ARP entry for each of these SNIPs. This is to ensure that the response traffic will reach the same interface through which the request went out. For more information on adding an ARP entry, see Configuring Static ARP.

   Command Line Interface example

```
> add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
 Done
> add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
 Done
> add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
 Done
> add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
 Done
```

Monitoring the Bridge Table and Changing the Aging time

NetScaler appliance bridges frames on the basis of bridge table lookup of the destination MAC address and the VLAN ID. However, the appliance performs forwarding only when Layer 2 mode is enabled.

The bridge table is dynamically generated, but you can display it, modify the aging time for the bridge table, and view bridging statistics.

To display the bridge table by using NetScaler command line

At the NetScaler command prompt, type:

sh bridgetable

Example

```
> show bridgetable

      Ageing time for bridge table entries : 300 seconds

      MAC               Iface    VLAN
      ---               -----    ----
1)    00:d0:68:0b:58:da    1/1      1
2)    00:00:5e:00:02:21    1/1      1
3)    00:11:95:1d:87:40    1/1      1
4)    00:d0:68:07:8b:bf    1/1      1
5)    00:e0:81:01:13:5a    1/1      1
6)    00:d0:68:10:6d:7a    1/1      1
7)    00:30:48:90:fa:d2    1/1      1
8)    02:d0:68:15:fd:3d    1/1      1
9)    00:0d:88:24:5f:30    1/1      1
10)   00:21:55:24:b8:3f    1/1      1
11)   00:d0:68:15:fd:36    1/1      1
 Done
```

To display the bridge table by using the configuration utility

1. In the navigation pane, expand **Network** and click **Bridge Table**.
2. Optionally on the **Bridge Table** page, select an entry to display its properties at the bottom of the screen.

To change the aging time by using the NetScaler command line

At the NetScaler command prompt, type:

- set bridgetable -bridgeAge <positive_integer>
- show bridgetable

Example

```
> set bridgetable -bridgeage 70
 Done
> show bridgetable

      Ageing time for bridge table entries : 70 seconds

      MAC               Iface    VLAN
      ---               -----    ----
1)     00:d0:68:0b:58:da    1/1      1
2)     00:00:5e:00:02:21    1/1      1
3)     00:11:95:1d:87:40    1/1      1
4)     00:d0:68:07:8b:bf    1/1      1
5)     00:e0:81:01:13:5a    1/1      1
6)     00:d0:68:10:6d:7a    1/1      1
7)     00:30:48:67:11:00    1/1      1
8)     00:30:48:90:fa:d2    1/1      1
9)     02:d0:68:15:fd:3d    1/1      1
10)    00:0d:88:24:5f:30    1/1      1
11)    00:21:55:24:b8:3f    1/1      1
12)    00:d0:68:15:fd:36    1/1      1
 Done
```

Parameter for changing the aging time

bridgeAge

The bridge aging time in seconds. Possible values: 60 to 300. Default: 300.

To change the aging time by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Bridge Table**.
2. In the details pane, click **Change Ageing Time**.
3. In the **Change Ageing Time** dialog box, in the **Ageing Time (seconds)** text box, type the aging time (for example, **70**).
4. Click **OK**. All the MAC entries in the bridge table are updated with the aging time.

To view the statistics of a bridge table by using the NetScaler command line

At the NetScaler command prompt, type:

stat bridge

Example

```
> stat bridge
Bridging Statistics
                    Rate (/s)          Total
Loops                    0                 0
Collisions               0                 0
Interface muted          0                 0
 Done
```

To view the statistics of a bridge table by using the configuration utility

1. On the **Bridge Table** page, select the MAC address for which you want to view the statistics (for example, **00:12:01:0a:5f:46**).
2. Click **Statistics**.

Understanding NetScaler Appliances in Active-Active Mode Using VRRP

An active-active deployment, in addition to preventing downtime, makes efficient use of all the NetScaler appliances in the deployment. In active-active deployment mode, the same VIPs are configured on all NetScaler appliances in the configuration, but with different priorities, so that a given VIP can be active on only one appliance at a time.

Note: This feature is supported only on NetScaler nCore builds.

The active VIP is called the master VIP, and the corresponding VIPs on the other NetScaler appliances are called the backup VIPs. If a master VIP fails, the backup VIP with the highest priority takes over and becomes the master VIP. All the NetScaler appliances in an active-active deployment use the Virtual Router Redundancy Protocol (VRRP) protocol to advertise their VIPs and the corresponding priorities at regular intervals.

NetScaler appliances in active-active mode can be configured so that no NetScaler is idle. In this configuration, different sets of VIPs are active on each NetScaler. For example, in the following diagram, VIP1, VIP2, VIP3, and VIP4 are configured on appliances NS1, NS2, and NS3. Because of their priorities, VIP1 and VIP 2 are active on NS1, VIP3 is active on NS2 and VIP 4 is active on NS3. If, for example, NS1 fails, VIP1 on NS3 and VIP2 on NS2 become active.

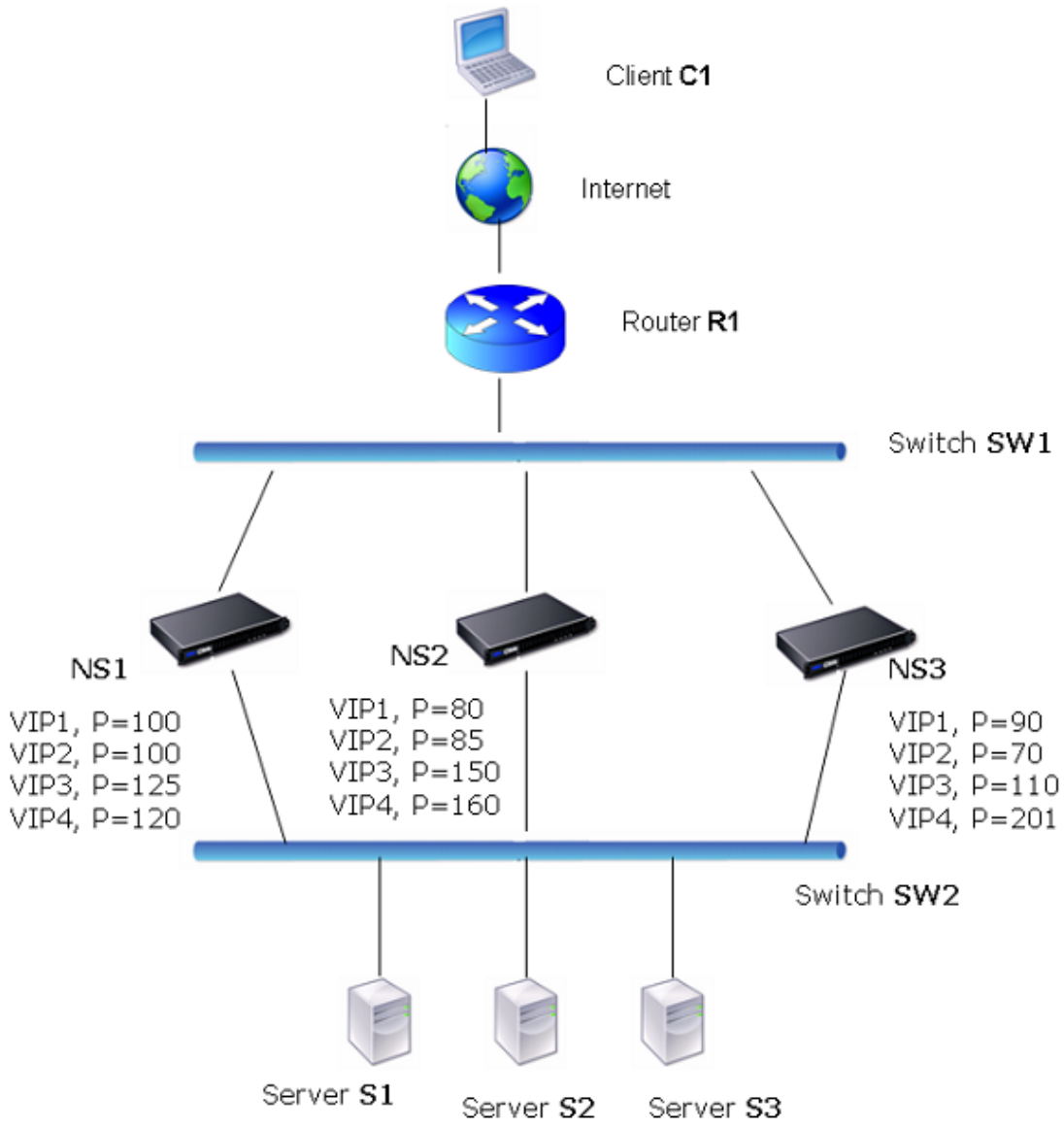


Figure 1. An Active-Active Configuration

The NetScaler appliances in the above diagram process traffic as follows:

1. Client C1 sends a request to VIP1. The request reaches R1.
2. R1 does not have an APR entry for VIP1, so it broadcasts an ARP request for VIP1.
3. VIP1 is active in NS1, so NS1 replies with a source MAC address as the VMAC (for example VMAC1) associated with VIP1, and VIP1 as the source IP address.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table.
5. R1 updates the ARP entry with VMAC1 and VIP1.
6. R1 forwards the packet to the VIP1 on NS1.

7. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP or MIP addresses and S2.

8. S2 replies to the SNIP or MIP on the NetScaler.

9. NS1 sends S2's reply to the client. In the reply, NS1 inserts MAC address of the physical interface as the source MAC address and VIP1 as the source IP address.

10. Should NS1 fail, the NetScaler appliances use the VRRP protocol to select the VIP1 with the highest priority. In this case, VIP1 on NS3 becomes active, and the following two steps update the active-active configuration.

11. NS3 broadcasts a GARP message for VIP1. In the message, VMAC1 is the source MAC address and VIP1 is the source IP address.

12. SW1 learns the new port for VMAC1 from the GARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS3. R1 updates its ARP table.

The priority of a VIP can be modified by health tracking. If you enable health tracking, you should make sure that preemption is also enabled, so that a VIP whose priority is lowered can be preempted by another VIP.

In some situations, traffic might reach a backup VIP. To avoid dropping such traffic, you can enable sharing, on a per-node basis, as you create an active-active configuration. Or you can enable the global send to master option. On a node on which sharing is enabled, it takes precedence over send to master.

Health Tracking

Base priority (BP-range 1-255) ordinarily determines which VIP is the master VIP, but effective priority (EP) can also affect the determination.

For example, if a VIP on NS1 has a priority of 101 and same VIP on NS2 has a priority of 99, the VIP on NS1 is active. However, if two vservers are using the VIP on NS1 and one of them goes DOWN, health tracking can reduce the EP of VIP on NS1. VRRP then makes the VIP on NS2 the active VIP.

Following are the health tracking options for modifying EP:

- **NONE.** No tracking. EP = BP

- **ALL.** If all virtual servers are UP, then EP = BP. Otherwise, EP = 0.

- **ONE.** If at least one virtual server is UP, then EP = BP. Otherwise, EP = 0.

- **PROGRESSIVE.** If ALL virtual servers are UP, then EP = BP. If ALL virtual servers are DOWN then EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP and k is the number of virtual servers that are down.

Note: If you specify a value other than NONE, preemption should be enabled, so that the backup VIP with the highest priority becomes active if the priority of the master VIP is downgraded.

Preemption

Preemption of an active VIP by another VIP that attains a higher priority is enabled by default, and normally should be enabled. In some cases, however, you may want to disable it. Preemption is a per-node setting for each VIP.

Preemption can occur in the following situations:

- An active VIP goes down and a VIP with a lower priority takes its place. If the VIP with the higher priority comes back online, it preempts the currently active VIP.
- Health tracking causes the priority of a backup VIP to become higher than that of the active VIP. The backup VIP then preempts the active VIP.

Sharing

In the event that traffic reaches a backup VIP, the traffic is dropped unless the sharing option is enabled on the backup VIP. This behavior is a per node setting for each VIP and is disabled by default.

In the figureAn Active-Active Configuration, VIP1 on NS1 is active and VIP1 VIPs on NS2 and NS3 are backups. Under certain circumstances, traffic may reach VIP1 on NS2. If Sharing is enabled on NS2, this traffic is processed instead of dropped.

Configuring Active-Active Mode

On each NetScaler appliance that you want to deploy in active-active mode, you must add a VMAC and bind the VMAC to a VIP. The VMAC for a given VIP must be same on each appliance. For example, if VIP 10.102.29.5, is created on the appliances, a virtual router ID must be created on each NetScaler and bound to VIP 10.102.29.5 on each NetScaler. When you bind a VMAC to a VIP, the NetScaler sends VRRP advertisements to each VLAN that is bound to that VIP. The VMAC can be shared by different VIPs configured on the same NetScaler.

Adding a VMAC

To add a VMAC for an active-active configuration, you create a virtual router ID. To bind a VMAC to a VIP, you associate the VMAC's virtual router ID with the VIP.

To add a VMAC by using the NetScaler command line

At the NetScaler command prompt, type:

add vrID <value> **-priority** <value> **-preemption** (ENABLED|DISABLED) **-sharing** (ENABLED | DISABLED) **-tracking** (NONE|ONE|ALL|PROGRESSIVE)

Example

```
add vrID 125 -priority 100 -sharing ENABLED -tracking ONE
```

Parameters for configuring a VMAC

vrID

The VRID that identifies the VMAC. Possible values: 1 - 255.

priority

The base priority of the VMAC. Range: 1 - 255. Default: 255.

tracking

The health tracking options for this VMAC. Possible values: NONE, ONE, ALL, PROGRESSIVE Default: NONE.

preemption

Make a backup VIP the master if its priority becomes higher than that of a master VIP that is bound to this VMAC. Possible values: ENABLED, DISABLED. Default: ENABLED.

sharing

Enable or disable sharing for this VMAC. Default: Disabled.

To add a VMAC by using the configuration utility

1. In the navigation pane, expand **Network** and click **VMAC**.
2. On the **VMAC** page, click **Add**.
3. In the **Add VMAC** dialog box, in **Virtual Router ID** text box, type a number (for example, **125**) to assign as the VMAC ID.
4. In the **Priority** text box, enter a priority number (for example, **100**) that will associated with VIPs bound this VMAC.
5. In the **Tracking** drop down box, select a health tracking option (for example, **ONE**).
6. Select or clear the **Preemption** check box to disable or enable preemption on VIPs that are bound to this VMAC.
7. Select or clear the **Sharing** check box to enable or disable sharing on VIPs that are bound to this VMAC.
8. Click **Create**.

To bind a VMAC by using the NetScaler command line

At the NetScaler command prompt, type:

set ns ip VIP <address> **-vrid** <value>

Example

set ns ip 10.102.29.5 -vrid 125

To bind a VMAC to a VIP by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **IPs**.
2. In the details pane, on the **IPv4s** tab, select the VIP address (for example, **10.102.29.5**) that you want to bind to a VMAC, and then click **Open**.
3. In the **Configure IP** dialog box, in the **Virtual Router Id** drop down box, select a virtual router ID (for example, **125**).
4. Click **OK**.

Configuring Send to Master

Usually, the traffic destined to a VIP reaches the NetScaler appliance on which the VIP is active, because an ARP request with the VIP and a VMAC on that appliance has reached the upstream router. But in some cases, such as static routes configured on the upstream router for the VIP subnet, or a topology that blocks this route, the traffic can reach a NetScaler appliance on which the VIP is in backup state. If you want this appliance to forward the data packets to the appliance on which the VIP is active, you need to enable the send to master option. This behavior is a per node setting and is disabled by default.

For example, in the following diagram, VIP1 is configured on NS1, NS2, and NS3 and is active on NS1. Under certain circumstances, traffic for VIP1 (active on NS1) may reach VIP1 on NS3. When the send to master option is enabled on NS3, NS3 forwards the traffic to NS1 through NS2 by using route entries for NS1.

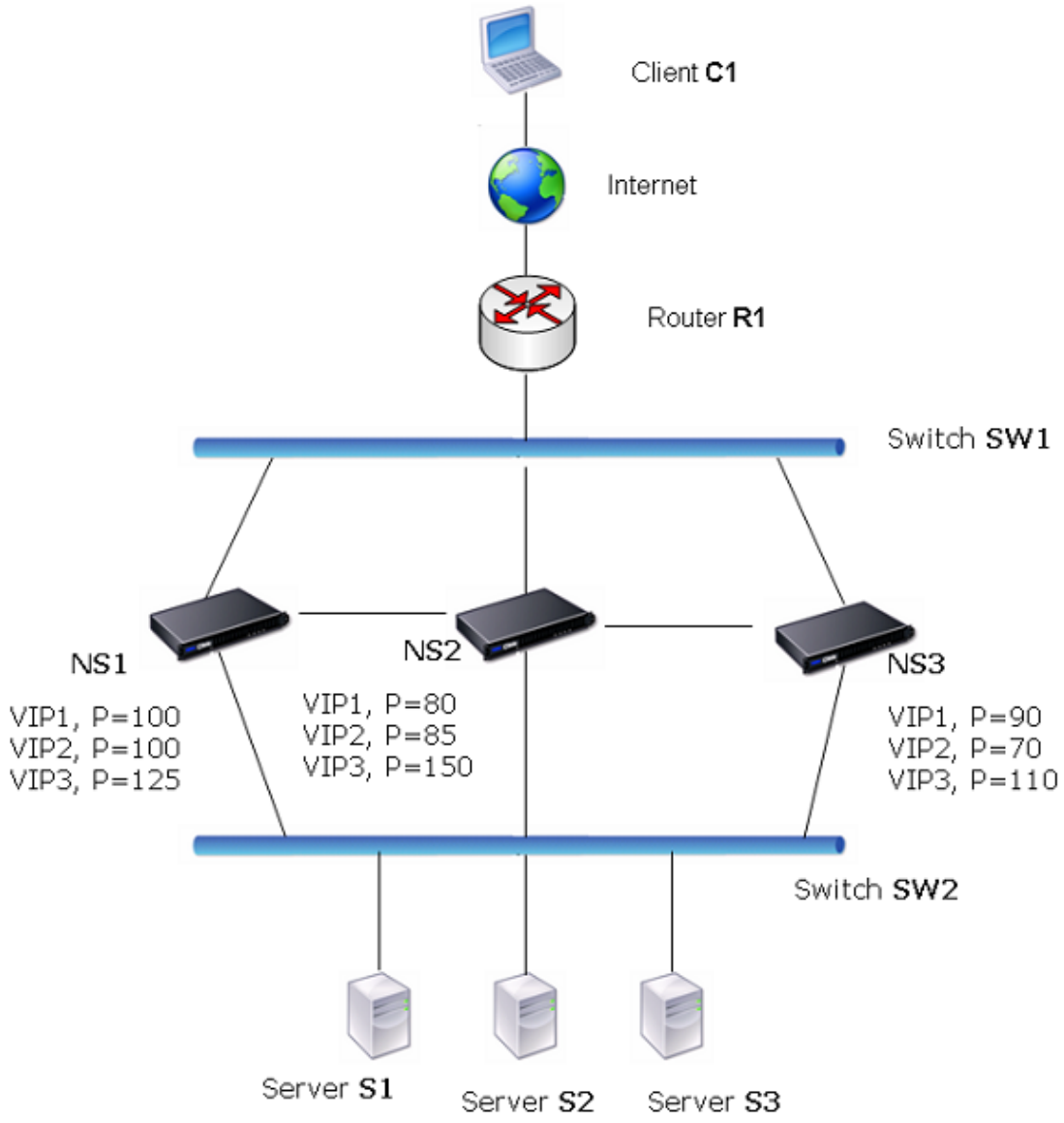


Figure 1. An Active-Active Configuration with Send to Master Option Enabled

To enable send to master by using the NetScaler command line

At the NetScaler command prompt, type:

set vrIDParam **-sendToMaster** (ENABLED|DISABLED)

Example

```
set vrIDParam -sendToMaster ENABLED
```

Parameter for enabling send to master

sendToMaster

Forward the packet to the master node if the VIP bound to the VMAC is in backup state and sharing is disabled.

Possible values: ENABLED, DISABLED. Default: DISABLED.

To enable send to master by using the configuration utility

1. In the navigation pane, expand **Network**.
2. In the details pane, under **Settings**, click **Virtual Router Parameters**.
3. In the **Virtual Router Parameters** dialog box, select **Send to Master** option.
4. Click **OK**.

An Active-Active Deployment Scenario

Following is an example of a possible active-active deployment scenario.

In the following diagram, VIP1, VIP 2 and VIP3 are configured on all three appliances, NS1, NS2, and NS3. Base Priorities for each VIPs are as shown in the diagram. Health tracking is disabled for each VIP. The priorities of VIPs are set so that VIP1, VIP2, and VIP3 are active on NS3. If NS3 fails, VIP1, VIP2, and VIP3 become active on NS1.

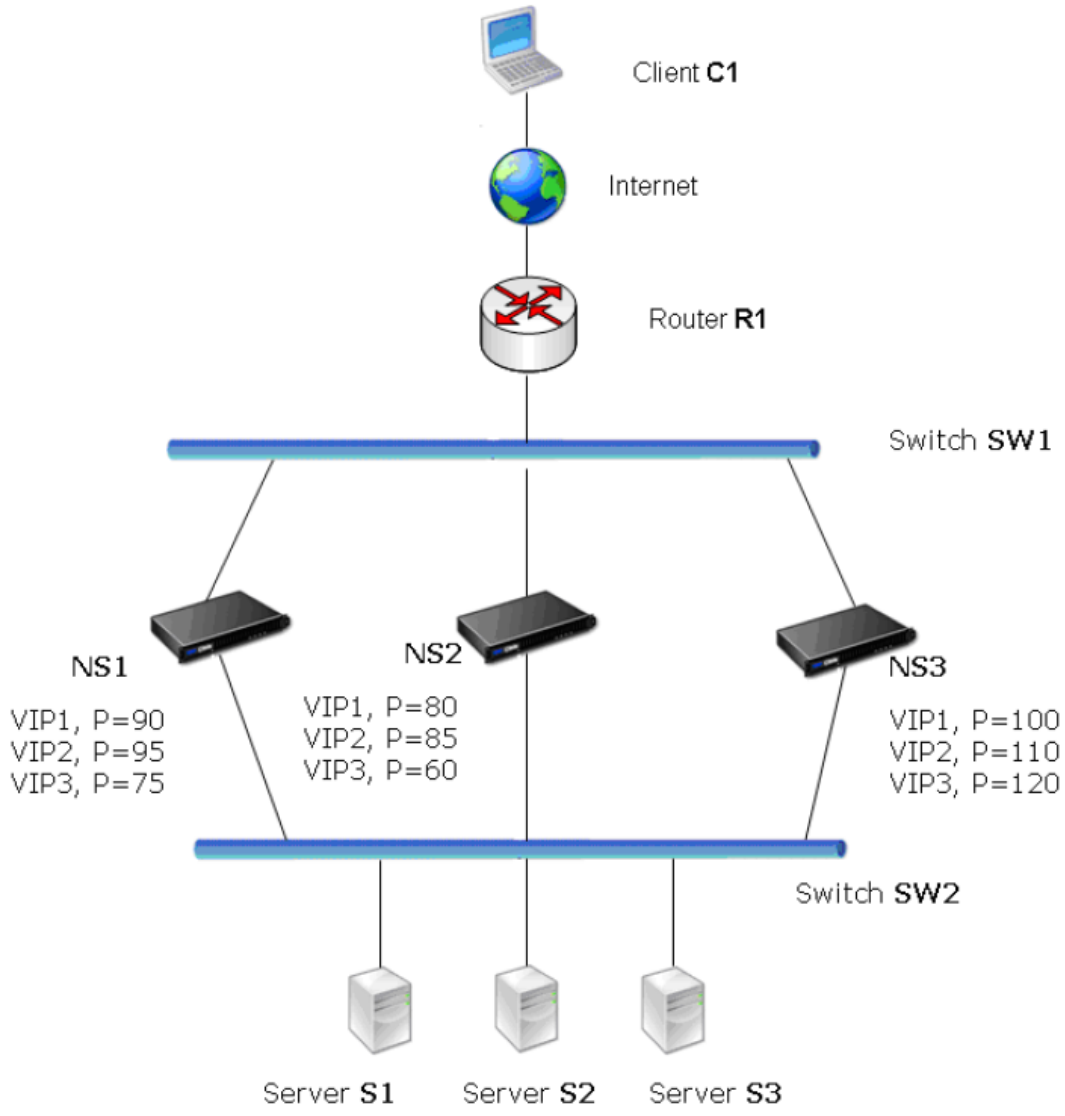


Figure 1. An Active-Active Deployment Scenario

Using the Network Visualizer

The Network Visualizer is a tool that you can use to view the network configuration of a NetScaler node, including the network configuration of the nodes in a high availability (HA) deployment. You can also modify the configuration of VLANs, interfaces, channels, and bridge groups, and perform HA configuration tasks.

In an HA deployment, you can both view and configure network entities on the node to which you are logged on, but you can view the details of only the network entities that are configured on the peer node. However, you can perform certain tasks, such as viewing details and statistics of the peer node and forcing a failover.

When you are logged on to a standalone appliance, you can use the Network Visualizer to do the following:

- View a consolidated graphical summary of key network components, such as VLANs, interfaces, channels, and bridge groups. You can also view the individual details of various network components.
- Modify appliance settings.
- Add, modify, and enable and disable interfaces and channels that are configured on the NetScaler appliance.
- Add and modify VLANs and bridge groups.
- Configure an HA deployment (add a node).
- View node details, node statistics, and statistics for VLANs and interfaces.
- Copy the properties of a network entity to a document or spreadsheet.

When you are logged on to an appliance in an HA deployment, you can perform the above tasks only on the appliance to which you are logged on. Following are additional tasks that you can perform in the Network Visualizer when you are logged on to one of the appliances in an HA pair:

- View the configuration details and high availability details of both nodes in an HA pair.
- Perform HA configuration tasks, such as synchronization and force failover.
- Remove the peer node from the HA configuration.
- View statistics for the peer node.
- Copy the properties of the peer node to a document or spreadsheet.

To open the Network Visualizer

1. In the navigation pane, click **Network**.
2. In **Monitor Connections**, click **Network Visualizer**.

To locate a VLAN or bridge group in the Visualizer

1. Open the **Network Visualizer**, and then do the following:
   - To locate a VLAN or bridge group, in the **Search** text field, begin typing the ID of the VLAN or the bridge group that you want to locate.

     Alternatively, begin typing the IP address of a bound subnet or the ID of a bound interface. The VLANs or bridge groups whose names match the typed characters are highlighted.

     To highlight multiple entities simultaneously, separate the IDs and IP addresses with white spaces. Entities whose IDs or IP addresses match any of the typed IDs and IP addresses are highlighted.
   - To clear the Search field, click the **x** adjacent to the field.

To view the configuration details of an entity by using the Visualizer

1. Open the **Network Visualizer** and do one of the following:
   - To view a brief summary of the entity, place the pointer on the entity.

     A brief summary of the entity appears at the bottom of the viewable area.
   - To view the detailed configuration information about the entity, click the entity.

     The configuration details for that entity appear in the Details area.

To modify the network settings of the appliance by using the Visualizer

1. Open the **Network Visualizer** and click the icon representing the appliance to which you are logged on.
2. In **Related Tasks**, click **Open**.

To add a channel by using the Visualizer

1. Open the **Network Visualizer** and click a network interface.
2. In **Related Tasks**, click **Add Channel**.

To add a VLAN by using the Visualizer

1. Open the **Network Visualizer**, click the appliance to which you are logged on, and then do one of the following:
   - Click an existing **VLAN**, and then, in **Related Tasks**, click **Add**.
   - Click an existing bridge group, and then, in **Related Tasks**, click **Add VLAN**.

To add a bridge group by using the Visualizer

1. Open the **Network Visualizer**, click the appliance to which you are logged on, and then do one of the following:
   - Click an existing bridge group, and then, in **Related Tasks**, click **Add**.
   - Click an existing **VLAN**, and then, in **Related Tasks**, click **Add Bridge Group**.

To modify the settings of an interface or channel by using the Visualizer

1. Open the **Network Visualizer** and click the interface whose settings you want to modify.
2. In **Related Tasks**, click **Open**.

To enable or disable an interface or channel by using the Visualizer

1. Open the **Network Visualizer** and click the interface or channel that you want to enable or disable.
2. In **Related Tasks**, do one of the following.
   - To enable the interface or channel, click **Enable**.
   - To disable the interface or channel, click **Disable**.

To remove a configured channel, VLAN, or bridge group by using the Visualizer

1. Open the **Network Visualizer** and click the channel, VLAN, or bridge group that you want to remove from the configuration.
2. In **Related Tasks**, click **Remove**.

To view statistics for a node, channel, interface, or VLAN by using the Visualizer

1. Open the **Network Visualizer** and click the node, interface, or VLAN whose statistics you want to view.
2. In **Related Tasks**, click **Statistics**.

To set up an HA deployment by using the Visualizer

1. Open the **Network Visualizer** and click the appliance.
2. In **Related Tasks**, click **HA Setup**.

To view the high availability details of a node by using the Visualizer

1. Open the **Network Visualizer** and click the node whose high availability details you want to view.
2. In **Related Tasks**, click **Details**.

To force the secondary node to take over as the primary by using the Visualizer

1. Open the **Network Visualizer** and click one of the nodes.
2. In **Related Tasks**, click **Force Failover**.

To synchronize the secondary node's configuration with the primary node by using the Visualizer

1. Open the **Network Visualizer** and click one of the nodes.
2. In **Related Tasks**, click **Force Synchronization**.

To remove the peer node from the HA configuration

1. Open the **Network Visualizer** and click the peer node.
2. In **Related Tasks**, click **Remove**.

To copy the properties of a node or network entity by using the Visualizer

1. Open the **Network Visualizer** and click the appliance or network entity whose properties you want to copy to a document or spreadsheet.
2. In **Related Tasks**, click **Copy Properties**.

Access Control Lists

Access Control Lists (ACLs) filter IP traffic and secure your network from unauthorized access. An ACL consists of a set of conditions that the NetScaler® appliance uses to allow or deny access. Consider a small organization that consists of 3 departments, Finance, HR, and Documentation, where no department wants another to access its data. The administrator of the organization can configure ACLs on the NetScaler to allow or deny access. When the NetScaler receives a data packet, it compares the information in the data packet with the conditions specified in the ACL and allows or denies access. The NetScaler supports simple ACLs, extended ACLs, and ACL6s. If both simple and extended ACLs are configured, incoming packets are compared to the simple ACLs first.

Simple ACLs filter packets on the basis of their source IP address and, optionally, their destination port and/or their protocol. Any packet that has the characteristics specified in the ACL is dropped.

Extended ACLs filter data packets on the basis of various parameters, such as source IP address, source port, action, and protocol. An extended ACL defines the conditions that a packet must satisfy for the NetScaler to process the packet, bridge the packet, or drop the packet. These actions are known as "processing modes."

The processing modes are:

- ALLOW - The NetScaler processes the packet.
- BRIDGE - The NetScaler bridges the packet to the destination without processing it.
- DENY - The NetScaler drops the packet.

The NetScaler processes an IP packet directly when both of the following conditions exist:

- ACLs are configured on the NetScaler.
- The IP packet does not match any of the ACLs

ACL6s are ACLs created specifically for IPv6 addresses. ACL6s filter packets on the basis of packet parameters, such as source IP address, source port, action, and so on. An ACL6 defines the condition that a packet must satisfy for the NetScaler to process the packet, bridge the packet, or drop the packet. These actions are known as "processing modes."

The processing modes are:

- ALLOW - The NetScaler processes the packet.
- BRIDGE - The NetScaler bridges the packet to the destination without processing it.
- DENY - The NetScaler drops the packet.

The NetScaler processes an IP packet directly when both of the following conditions exist:

- ACL6s are configured on the NetScaler.
- The IP packet does not match any of the ACL6s.

ACL Precedence

An IPv4 packet that matches the conditions specified in a simple ACL is dropped. If the packet does not match any simple ACL, the NetScaler compares the packet's characteristics to those specified in any configured extended ACLs. If the packet matches an extended ACL, the NetScaler applies the action specified in theExtended ACL, as shown in the following diagram.

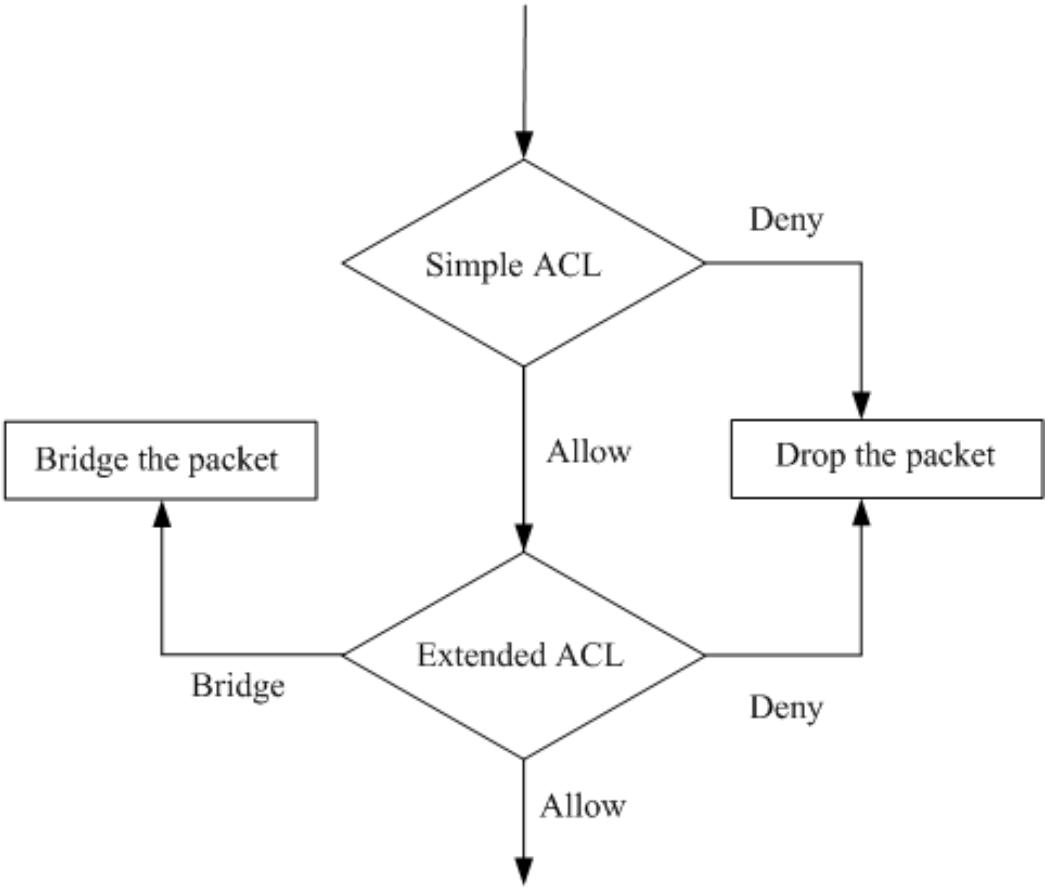


Figure 1. Simple and Extended ACLs Flow Sequence

IPv6 packets are compared only to ACL6s.

Configuring Simple ACLs

A simple ACL, which uses few parameters, cannot be modified once created. When creating a simple ACL, you can specify a time to live (TTL), in seconds, after which the ACL expires. ACLs with TTLs are not saved when you save the configuration. You can also remove a simple ACL manually. You can display simple ACLs to verify their configuration, and you can display statistics to monitor their performance.

Creating Simple ACLs

Use either of the following procedures to create a simple ACL.

To create a simple ACL by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an ACL and verify the configuration:

- add ns simpleacl <aclname> **DENY -srcIP** <ip_addr> [**-destPort**<port> **-protocol** (TCP | UDP)] [**-TTL** <positive_integer>]
- show ns simpleacl [<aclname>]

Example

```
> add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
 Done

> show simpleacl rule1
      Name: rule1                    Action: DENY
      srcIP = 10.102.29.5
      Protocol:                     DestPort:
      Hits: 0                      TTL: 590(seconds)
 Done
```

Parameters for configuring a Simple ACL

Parameter

Specifies

aclName

Alphanumeric name of the ACL. Maximum length: 127 characters.

srcIP

IP address of the source machine. You can also specify a range of addresses.

destPort

A destination port on the NetScaler. If you do not specify a port, you create an all-ports ACL, which matches any port. In that case, you cannot create another ACL specifying a specific port and the same source IP address.

protocol

Underlying protocol for this connection. You must specify a value for this parameter if you set the destPort parameter. Possible values: TCP or UDP.

TTL

Number of seconds after which the ACL is to expire. Possible values: 1 to 2147483647. Default: The ACL does not expire. (If you do not want the ACL to expire, do not specify a TTL value.)

To create a simple ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the **ACLs** pane, on the **Simple ACLs** tab, click **Add**.
3. In the **Add Simple ACL** dialog box, specify values for the following parameters:
   - **Name*—**aclName
   - **Protocol—**protocol
   - **Source IP Address—**srcIP
   - **Destination Port**—port
   - **TTL**—TTL (If you do not want the ACL to expire, leave the TTL field blank.)

   *A required parameter
4. Click **Create,** and then click **Close.**
5. On the **Simple ACLs** tab, select the ACL that you created and verify that the settings displayed at the bottom of the screen are correct.

Monitoring Simple ACLs

You can display the simple-ACL statistics, which include the number of hits, the number of misses, and the number of simple ACLs configured.

To view simple ACL statistics by using the NetScaler command line

At the NetScaler command prompt, type:

stat ns simpleacl

Example

```
>stat ns simpleacl

                        Rate (/s)          Total
Deny SimpleACL hits             0              0
SimpleACL hits                  0              0
SimpleACL misses                0             11
SimpleACLs count               --              1
 Done
```

The following table describes statistics you can display for simple ACLs.

Table 1. Simple ACL Statistics

Statistic	Indicates
Deny SimpleACL hits	Packets dropped because they match deny simple ACL
SimpleACL hits	Packets matching a simple ACL
SimpleACL misses	Packets not matching any simple ACL
SimpleACL count	Number of simple ACLs configured

To display simple ACL statistics by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, select the **ACL** whose statistics you want to display (for example, rule1).
3. Click **Statistics**.
4. View the ACL statistics in the new window that opens.

Removing Simple ACLs

If you need modify a simple ACL, you must remove it and create a new one.

To remove a single simple ACL by using the NetScaler command line

At the NetScaler command prompt, type:

- rm ns simpleacl <aclname>
- show ns simpleacl

To remove all simple ACLs by using the NetScaler command line

At the NetScaler command prompt, type:

- clear ns simpleacl
- show ns simpleacl

To remove a single simple ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Simple ACLs** tab, select the simple ACL that you want to remove (for example, rule1).
3. Click **Remove**.
4. In the **Remove** dialog box, click **Yes**.
5. In the details pane, on the **Simple ACLs** tab, verify that the entry for rule1 has been removed

To remove all simple ACLs by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Simple ACLs** tab, click **Clear**.
3. In the **Clear Simple ACL (s)** dialog box, click **Yes**.
4. In the details pane, verify that there are no entries in the **Simple ACLs** tab.

Configuring Extended ACLs

To configure extended ACLs, many users first create extended ACLs and then modify them.

For any of the following actions to take effect, they must be applied, by clicking the **Commit** button:

- Activate
- Remove
- Disable
- Change the Priority

Other actions include:

- Configure logging
- Verify the configuration
- Monitor ACL statistics

Note: If you configure both simple and extended ACLs, simple ACLs take precedence over extended ACLs.

Parameters of Extended ACLs can be configured during creation. Additionally, the following actions can be performed on Extended ACLs: Modify, Remove, Apply, Disable, Enable and Renumber the priority of Extended ACLs.

You can collect statistics of packets using Extended ACLs by enabling logging.

Creating and Modifying an Extended ACL

To create an extended ACL by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns acl <aclname> <aclaction> [**-srcIP** [<operator>] <srcIPVal>] [**-srcPort** [<operator>] <srcPortVal>] [**-destIP** [<operator>] <destIPVal>] [**-destPort** [<operator>] <destPortVal>] [**-TTL** <positive_integer>] [**-srcMac** <mac_addr>] [(**-protocol** <protocol> [**-established**]) | **-protocolNumber** <positive_integer>] [**-vlan** <positive_integer>] [**-interface** <interface_name>] [**-icmpType** <positive_integer> [**-icmpCode** <positive_integer>]] [**-priority** <positive_integer>] [**-state** (ENABLED | DISABLED)] [**-logstate** (ENABLED | DISABLED) [**-ratelimit** <positive_integer>]]

- show ns acl [<aclname>]

Example

```
> add ns acl restrict DENY -srcport 45-1024 -destIP 192.168.1.1 -protocol TCP
 Done
> show ns acl restrict
      Name: restrict                  Action: DENY     Hits: 0
      srcIP
      destIP = 192.168.1.1
      srcMac:                       Protocol: TCP
      srcPort =  45-1024               destPort
      Vlan:                        Interface:
      Active Status: ENABLED             Applied Status: NOTAPPLIED
      Priority: 10                   NAT: NO
      TTL:
      Log Status: DISABLED
 Done
```

Parameters for configuring an extended ACL

aclname

Alphanumeric name of the ACL. Maximum length: 127 characters.

aclaction

The action associated with the ACL. The valid options for this parameter are BRIDGE, DENY, and ALLOW.

srcIP

IP address of the source machine. You can also specify a range of addresses, by enclosing the low and high addresses in brackets (for example, [10.102.29.30-10.102.29.189]).

operator

You can use the following operators while creating ACLs: = and !=.

destIP

The IP address of the destination system. You can also specify a range of addresses, by enclosing the low and high addresses in brackets (for example, [10.102.33.31-10.102.33.100]).

protocol

The protocol field in the IP header. Possible values: ICMP, IGMP, TCP, EGP, IGP, ARGUS, UDP, RDP, RSVP, EIGRP, L2TP, and ISIS.

protocolNumber

The IP protocol number (decimal). Minimum value: 1. Maximum value: 255.

srcPort

The port address of the source system. You can also specify a range of ports, by enclosing the low and high port numbers in brackets (for example [30-90]).

Note: The Source Port can be modified only for TCP and UDP.

destPort

The port address of the destination system. You also can specify a range of ports, by enclosing the low and high port numbers in brackets (for example [30-90]).

Note: The Destination Port can be modified only for TCP and UDP.

established

Use the ACL for TCP response traffic only.

TTL

ACLs can be configured to expire after a specified amount of time (in seconds). Possible values: 1 to 2147483647. Default: The ACL does not expire. (If you do not want the ACL to expire, do not specify a TTL value.)

srcMac

The MAC address of the source system. Only the last 32 bits are considered during a lookup.

vlan

The VLAN ID present in the VLAN tag of the packet. Possible values: 1 to 4094.

interface

This is the network interface on which the packet arrived.

icmpType

The ICMP message type. For example, to block DESTINATION UNREACHABLE messages, you must specify 3 as the ICMP type. For a complete list of ICMP types, see http://www.iana.org/assignments/icmp-parameters. Possible values: 0 to 255.

icmpCode

The ICMP message code. For example, to block DESTINATION HOST UNREACHABLE messages, specify 3 as the ICMP type and 1 as the ICMP code. For a complete list of ICMP types and codes, see http://www.iana.org/assignments/icmp-parameters. Possible values: 0 to 255.

priority

The priority of the ACL, which determines the order in which it will be evaluated relative to other extended ACLs. Possible values: 0 to 10240.

state

The state of the ACL. Possible values: ENABLED, DISABLED. Default: Enabled.

ratelimit

Log message rate limit for ACL. Possible values: 1 to 10000. Default:100.

To create an extended ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Extended ACLs** tab, click **Add**.
3. In the **Create ACL** window, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an extended ACL" as shown:

   - **Name***—aclname
   - **Action***—aclaction
   - **Source**, **Operator**—operator
   - **Source**, **Low/High**—srcIP (To specify a single IP address, type the same address in both fields.)
   - **Destination**, **Operator**—operator
   - **Destination**, **Low/High**—destIP (To specify a single IP address, type the same address in both fields.)
   - **Protocol**—protocol
   - **Source Port, operator**—operator
   - **Source Port**, **Low/High**—srcPort (To specify a single port, type the same port number in both fields.)
   - **Destination Port, Operator**—operator
   - **Destination Port, Low/High**—destPort (To specify a single port, type the same port number in both fields.)
   - **Established**—established
   - **ICMP Message Type**—icmpType
   - **ICMP Message Code**—icmpCode
   - **Source Mac**—srcMac
   - **VLAN**—vlan
   - **Interface**—interface
   - **Priority**—priority
   - **TTL**—TTL

- **Enable ACL**—state
- **Log State**—logstate
- **Log Rate Limit**—ratelimit

*A required parameter

4. Click **Create**, and then click **Close**.
5. In the details pane, verify that the settings for ACL that you configured are correct.

Applying an Extended ACL

After you create or modify an extended ACL, you must activate it by using one of the following procedures. These procedures reapply all the ACLs.

For example, if you have created the ACLs rule1 through rule10, and then you create an ACL called rule11, and apply it, all of the ACLs (rule1 through rule11) are applied afresh.

If a session has a DENY ACL related to it, that session is destroyed.

To apply an ACL by using the NetScaler command line

At the NetScaler command prompt, type:

- apply ns acls
- show ns acl

To apply an ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. Click **Commit**.
3. In the **Apply ACL(s)** dialog box, click **Yes**.
4. Verify the information on the **Extended ACLs** tab.

Disabling and Enabling Extended ACLs

By default, ACLs are enabled. This means when ACLs are applied, the NetScaler appliance compares incoming packets against the ACLs.

Disable an ACL if it will not be used for a certain period. After the ACLs are applied, the NetScaler does not compare incoming packets against disabled ACLs.

To disable or enable an extended ACL by using the NetScaler command line

At the NetScaler command prompt, type one of the following pairs of commands to disable or enable an ACL and verify the result:

- disable ns acl <aclname>
- show ns acl [<aclname>]
- enable ns acl <aclname>
- show ns acl [<aclname>]

Example

```
> disable ns acl restrict
 Done

> show ns acl restrict
      Name: restrict                  Action: DENY     Hits: 0
      srcIP
      destIP = 192.168.1.1
      srcMac:                       Protocol: TCP
      srcPort =  45-1024               destPort
      Vlan:                        Interface:
      Active Status: DISABLED            Applied Status: NOTAPPLIED
      Priority: 10                   NAT: NO
      TTL:
      Log Status: DISABLED
 Done

> enable ns acl restrict
 Done

> show ns acl restrict
```

```
    Name: restrict                  Action: DENY     Hits: 0
    srcIP
    destIP = 192.168.1.1
    srcMac:                       Protocol: TCP
    srcPort =  45-1024              destPort
    Vlan:                       Interface:
    Active Status: ENABLED            Applied Status: APPLIED
    Priority: 10                  NAT: NO
    TTL:
    Log Status: DISABLED
Done
```

To disable or enable an extended ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Extended ACLs** tab, select the ACL (for example, rule1) and click **Open**.
3. In the **Configure ACL** dialog box, select the **Enable ACL** check box to enable, or clear the check box to disable, the ACL.
4. Click **OK**.
5. If you want to apply the new setting, which reapplies all ACLs, click **Commit**, and then, in the **Apply ACL(s)** dialog box, click **Yes**.
6. In the details pane, on the **Extended ACLs** tab, view the list to verify the changed status under the column **Active Status**.

Renumbering the priority of Extended ACLs

The renumber procedure resets the priorities of the ACLs to multiples of 10. The priority (an integer value) defines the order in which the NetScaler appliance evaluates ACLs. All priorities are multiples of 10, unless you configure a specific priority to an integer value. When you create an ACL without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the ACL, the NetScaler performs an action. If the packet does not match the condition defined by the ACL, the NetScaler compares the packet against the ACL with the next-highest priority.

Consider the following example. Two ACLs, rule1 and rule2, are automatically assigned priorities 20 and 30 when they are created. You need to add a third ACL, rule3, to be evaluated immediately after rule1. Rule3 must have a priority between 20 and 30. In this case, you can specify the priority as 25. Later, you can easily renumber the ACLs with priorities that are multiples of 10, without affecting the order in which the ACLs are applied.

To renumber the ACLs by using the NetScaler command line

At the NetScaler command prompt, type:

renumber ns acls

To renumber the ACLs by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **ACLs**.
2. In the details pane, on the **Extended ACLs** tab, click **Renumber Priority (s)**.
3. In the **Renumber Priority (s) ACL(s)** dialog box, click **Yes**.
4. In the details pane, on the **Extended ACLs** tab, verify the changed priority.

Configuring Extended ACL Logging

You can configure the NetScaler appliance to log details for packets that match an extended ACL. In addition to the ACL name, the logged details include packet-specific information such as the source and destination IP addresses. The information is stored either in the syslog file or in the nslog file, depending on the type of global logging (syslog or nslog) enabled.

Logging can be enabled at both the global level and the ACL level. The global setting takes precedence.

For more information about enabling logging globally, see Audit Logging.the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

To optimize logging, when multiple packets from the same flow match an ACL, only the first packet's details are logged, and the counter is incremented for every packet that belongs to the same flow. A flow is defined as a set of packets that have the same values for the following parameters:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Protocol

If the packet is not from the same flow, or if the time duration is beyond the mean time, a new flow is created. Mean time is the time during which packets of the same flow do not generate additional messages (although the counter is incremented).

Note: The total number of different flows that can be logged at any given time is limited to 10,000.

To configure ACL Logging by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure logging and verify the configuration:

- set ns acl <aclName> [**-logState** (ENABLED | DISABLED)] [**-rateLimit** <positive_integer>]
- show ns acl [<aclName>]

Example

```
>set ns acl restrict -logstate ENABLED -ratelimit 120

Warning: ACL modified, apply ACLs to activate change

      > apply ns acls
      Done

> show ns acl restrict
      Name: restrict                  Action: DENY     Hits: 0
      srcIP
      destIP = 192.168.1.1
      srcMac:                        Protocol: TCP
      srcPort =  45-1024                 destPort
      Vlan:                         Interface:
      Active Status: ENABLED              Applied Status: APPLIED
      Priority: 10                    NAT: NO
      TTL:
      Log Status: ENABLED                 Log Rate limit: 120
 Done
```

Logging parameters of an extended ACL

aclName

The alphanumeric name of the ACL.

logState

State of the logging feature for the ACL. Possible Values: Enabled, Disabled. Default: Disabled.

rateLimit

Number of log messages that a specific ACL can generate. Default: 100.

To configure ACL Logging by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, click the **Extended ACLs** tab, and then select the ACL for which you want to configure logging (for example, rule1).
3. Click **Open**.
4. In the **Configure ACL** dialog box, specify values for the following parameters, which correspond to parameters described in "Logging parameters of an extended ACL" as shown:
   - **Log State**—logState
   - **Log Rate Limit**—rateLimit
5. Click **OK**.
6. In the**ACL modified, apply ACLs to activate change** dialog box, click **OK**.
7. Select the modified ACL and, under **Details**, verify the log state.

Monitoring the Extended ACL

You can display statistics for monitoring the performance of an extended ACL.

To display the statistics of an extended ACL by using the NetScaler command line

At the NetScaler command prompt, type:

stat ns acl

Example

```
>stat ns acl rule1

ACL: rule1
                          Rate (/s)          Total
Hits for this ACL                 0              0
 Done
```

The following table lists the statistics associated with extended ACLs and their descriptions.

Table 1. Extended ACL Statistics

Statistic	Specifies
Allow ACL hits	Packets matching ACLs with processing mode set to ALLOW. NetScaler processes these packets.
NAT ACL hits	Packets matching a NAT ACL, resulting in a NAT session.
Deny ACL hits	Packets dropped because they match ACLs with processing mode set to DENY.
Bridge ACL hits	Packets matching a bridge ACL, which in transparent mode bypasses service processing.
ACL hits	Packets matching an ACL.
ACL misses	Packets not matching any ACL.

To display the statistics of an extended ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Extended ACLs** tab, select the ACL whose statistics you want to view (for example, rule1).
3. Click **Statistics**.
4. View the statistics in the new window that opens.

Removing Extended ACLs

You can remove a single extended ACL or all extended ACLs.

To remove a single extended ACL by using the NetScaler command line

At the NetScaler command prompt, type:

- rm ns acl <aclName>
- show ns acl

To remove all extended ACLs by using the NetScaler command line

At the NetScaler command prompt, type:

- clear ns acls
- show ns acl

To remove a single extended ACL by using the configuration utility

1. In the navigation pane, expand **Network** and click ACLs.
2. In the details pane, on the **Extended ACLs** tab, select the ACL that you want to remove (for example, rule1).
3. Click **Remove**.
4. In the **Remove** dialog box, click **Yes**.

To remove all extended ACLs by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Extended ACLs** tab, click **Clear**.
3. In the **Clear ACL (s)** dialog box, click **Yes**.
4. In the details pane, on the **Extended ACLs** tab, verify that no ACLs are listed.

Configuring Simple ACL6s

A simple ACL6, which uses few parameters, cannot be modified once created. Instead, you must remove the simple ACL6 and create a new one. When creating a simple ACL6, you must specify its name, and a source IP address value against which to match packets. Optionally, you can specify a destination port and a time to live (TTL) value. A TTL is the number of seconds after which the simple ACL6 expires. ACL6s with TTLs are not saved when you save the configuration.

Creating Simple ACL6s

To create a simple ACL6, you must specify its name and source IP address. You can also specify a destination port and time to live (TTL).

To create a simple ACL6 by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a simple ACL6 and verify the configuration:

- add ns simpleacl6 <aclname> **DENY -srcIPv6** <ipv6_addr|null> [**-destPort**<port> **-protocol** (TCP | UDP)] [**-TTL** <positive_integer>]
- show ns simpleacl6 [<aclname>]

Example

```
>  add ns simpleacl6 rule1 DENY –srcIPv6 3ffe:192:168:215::82 -destPort 80 -Protocol TCP -TTL 9000
 Done

> show simpleacl6 rule1
      Name: rule1
      Action: DENY                    Hits: 5
      srcIP6= 3ffe:192:168:215::82
      Protocol: TCP                   DestPort = 80
      TTL: 8922(seconds)
 Done
```

Parameters for configuring a simple ACL6

acl6name (Name)

A name for the simple ACL6. The name can begin with a letter, number, or the underscore symbol, and can consist of up to 127 letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed after the simple ACL6 has been created.)

DENY

Drop the packet. This is the only action available for a simple ACL6.

srcIPv6 (Source IP Address)

The IP address that the simple ACL6 rule compares to the address in the source IP address field of every incoming IPv6 packet.

destPort (Destination Port)

A destination port on the NetScaler appliance. If you do not specify a port, you create an all-ports ACL6, which matches any port. In that case, you cannot create another simple ACL6 specifying a specific port and the same source IPv6 address.

protocol (Protocol)

Underlying protocol name that the simple ACL6 rule will check in protocol name field of all the incoming IPv6 packets. You must specify a value for this parameter if you set the destPort parameter. Possible values: TCP, UDP. Simple ACL6s can traverse the extension headers (if present) of the incoming IPv6 packets to identify the protocol name.

TTL (TTL)

Number of seconds after which the ACL6 is to expire. Possible values: 1 to 2147483647. Default: The ACL6 does not expire. (If you do not want the ACL6 to expire, do not specify a TTL value.)

To create a simple ACL6 by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **ACLs**.
2. In the **ACLs** pane, on the **Simple ACL6s** tab, click **Add**.
3. In the **Add Simple ACL6** dialog box, specify values for the following parameters:
   - **Name***
   - **Protocol**
   - **Source IP Address**
   - **Destination Port**
   - **TTL**

   *A required parameter
4. Click **Create**, and then click **Close.**
5. On the **Simple ACL6s** tab, select the ACL that you created and verify that the settings displayed at the bottom of the screen are correct.

To remove a single simple ACL6 by using the NetScaler command line

At the NetScaler command prompt, type:

- rm ns simpleacl6 <aclname>
- show ns simpleacl6

To remove all simple ACL6s by using the NetScaler command line

At the NetScaler command prompt, type:

- clear ns simpleacl6
- show ns simpleacl6

To remove one or all simple ACL6s by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **ACLs**.
2. In the details pane, on the **Simple ACL6s** tab, do one of the following:
   - Select the simple ACL6 that you want to remove, and then click **Remove**.
   - To remove all simple ACL6s, click **Clear**.
3. In the **Proceed or Clear Simple ACL6(s)** dialog box, click **Yes**.
4. In the details pane, on the **Simple ACL6s** tab, verify that the entry or entries have been removed.

Monitoring Simple ACL6s

You can display the following simple ACL6 statistics:

Table 1. Simple ACL6 Statistics

Statistic	Indicates
Deny simpleACL6 hits	Packets dropped because they match a simple deny ACL6
Simple ACL6 hits	Packets matching a simple ACL6
Simple ACL6 misses	Packets not matching any simple ACL6
Simple ACL6 count	Number of simple ACL6s configured

To display simple ACL6 statistics by using the NetScaler command line

At the NetScaler command prompt, type:

stat ns simpleacl6

Example

```
>stat ns simpleacl6

                        Rate (/s)          Total
Deny SimpleACL6 hits            0              0
SimpleACL6 hits                 0              0
SimpleACL6 misses               0             11
SimpleACL6s count              --              1
```

Done

To display simple ACL6 statistics by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **Simple ACL6s** tab, select the simple ACL6 whose statistics you want to display.
3. Click **Statistics**.

Configuring ACL6s

ACL6s can be configured during creation. Additionally, the following actions can be performed on ACL6s: Modify, Apply, Disable, Enable, Renumber and Remove the priority of ACL6s.Log files of ACL6s can be configured to collect statistics of packets. If a packet matches the condition defined by the ACL6, the NetScaler performs an action. If the packet does not match the condition defined by the ACL6, the NetScaler compares the packet against the ACL6 with the next-highest priority.

Creating and Modifying ACL6s

To create an ACL6 by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns acl6 <acl6name> <acl6action> [**-srcIPv6** [<operator>] <srcIPv6Val>] [**-srcPort** [<operator>] <srcPortVal>] [**-destIPv6** [<operator>] <destIPv6Val>] [**-destPort** [<operator>] <destPortVal>] [**-TTL** <positive_integer>] [**-srcMac** <mac_addr>] [(**-protocol** <protocol> [**-established**]) | **-protocolNumber** <positive_integer>] [**-vlan** <positive_integer>] [**-interface** <interface_name>] [**-icmpType** <positive_integer> [**-icmpCode** <positive_integer>]] [**-priority** <positive_integer>] [**-state** (ENABLED | DISABLED)]

- show ns acl6 [<acl6name>]

Example

Example
> add ns acl6 rule6 DENY -srcport 45-1024 -destIPv6 2001::45 -protocol TCP
 Done

> show ns acl6 rule6
 Name: rule6 Action: DENY
 srcIPv6
 destIPv6 = 2001::45
 srcMac: Protocol: TCP
 srcPort = 45-1024 destPort
 Vlan: Interface:
 Active Status: ENABLED Applied Status: NOTAPPLIED
 Priority: 10 Hits: 0
 TTL:
 Done

To modify or remove an ACL6 by using the NetScaler command line

- To modify an ACL6, type the set ns ACL6 command, the name of the ACL6, and the parameters to be changed, with their new values.

- To remove an ACL6, type the rm ns ACL6 command and the name of the <entity>.

Parameters for configuring an ACL6

acl6name (Name)

A name for the simple ACL6. The name can begin with a letter, number, or the underscore symbol, and can consist of up to 127 letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed after the ACL6 has been created.)

acl6action (Action)

The action associated with the ACL6. Possible values: BRIDGE, DENY, ALLOW.

srcIPv6 (Source, Low/High)

The IP address that the ACL6 rule checks in source IP address field of all the incoming IPv6 packets. You can also specify a range of addresses, by enclosing the low and high addresses in brackets.

operator (Operator)

The type of operation for matching the ACL6 against packets. Possible values: = (equals), != (does not equal).

destIPv6 (Destination, Low/High)

The IP address that the ACL6 rule compares to the address in the destination IP address field of every incoming IPv6 packet. You can also specify a range of addresses, by enclosing the low and high addresses in brackets, with a hyphen between the two addresses.

protocol (Protocol)

The protocol field in the IP header. Possible values: TCP, UDP, ICMPv6. ACL6s can traverse through the extension headers (if present) of the incoming IPv6 packets to find out the protocol name.

protocolNumber

The IP protocol number (decimal). Minimum value: 1. Maximum value: 255.

srcPort (Source Port, Low/High)

The port number that the ACL6 rule compares to the port number in the source port field of every incoming IPv6 packet. You also can specify a range of ports, by enclosing the low and high port numbers in brackets, with a hyphen between the low and high port numbers (for example [40-90]).

Note: The Source Port can be modified only for TCP and UDP.

destPort (Destination Port, Low/High)

The port number that the ACL6 rule compares to the port number in the destination port field of every incoming IPv6 packet. You also can specify a range of ports, by enclosing

the low and high port numbers in brackets (for example [30-90]).

Note: The Destination Port can be modified only for TCP and UDP.

established

Use the ACL for TCP response traffic only.

TTL (TTL)

Expire the ACL6 after the specified amount of time (in seconds). Possible values: 1 to 2147483647. Default: The ACL does not expire. (If you do not want the ACL to expire, do not specify a TTL value.)

srcMac (Source MAC

The MAC address of the source system. Only the last 32 bits are considered during a lookup.

vlan (VLAN)

The VLAN ID present in the VLAN tag of the packet. Possible values: 1 to 4094.

interface (Interface)

The network interface on which the packet arrived.

icmpType (ICMP Message Type)

The ICMPv6 message type. For example, to block DESTINATION UNREACHABLE messages, you must specify 1 as the ICMP type. For a complete list of ICMP types, see http://www.iana.org/assignments/icmp-parameters. Possible values: 0 to 255.

icmpCode (ICMP Message Code)

The ICMPv6 message code. For example, to block DESTINATION UNREACHABLE NO ROUTE TO DESTINATION messages, specify 1 as the ICMP type and 0 as the ICMPv6 code . For a complete list of ICMP types and codes, see http://www.iana.org/assignments/icmp-parameters. Possible values: 0 to 255.

priority (Priority)

The priority of the ACL6, which determines the order in which it will be applied relative to other ACL6s. Possible values: 0 to 10240.

state (Enable ACL6)

The state of the ACL6. Possible values: ENABLED, DISABLED. Default: ENABLED.

To create an ACL6 by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the ACL6s tab, do one of the following:
   - To create a new ACL6, click **Add**.
   - To modify an existing ACL6, select the <entity>, and then click **Open**.
3. In the **Create ACL** window, set the following parameters:

   - **Name***
   - **Action***
   - **Source**, **Operator**
   - **Source**, **Low/High** (To specify a single IP address, type the same address in both fields.)
   - **Destination**, **Operator**
   - **Destination**, **Low/High** (To specify a single IP address, type the same address in both fields.)
   - **Protocol**
   - **Source Port, operator**
   - **Source Port**, **Low/High** (To specify a single port, type the same port number in both fields.)
   - **Destination Port, Operator**
   - **Destination Port, Low/High** (To specify a single port, type the same port number in both fields.)
   - **Established**
   - **ICMP Message Type**
   - **ICMP Message Code**
   - **Source Mac**
   - **VLAN**
   - **Interface**
   - **Priority**
   - **TTL**

- **Enable ACL6**

*A required parameter

4. Click **Create** and click **Close**.
5. In the details pane, you can view the ACL you created under the **ACL6s** tab.

Applying ACL6s

After you create an ACL6, you must activate it. The following procedures reapply all the ACL6s.

For example, if you have created the ACL6s rule1 through rule10, and then you create an ACL6 called rule11 and apply it, all of the ACL6s (rule1 through rule11) are applied afresh.

If a session has a DENY ACL related to it, the session is destroyed.

You must apply one of the following procedures after every action you perform on an ACL6 (for example, after disabling an ACL6). However, you can add or modify more than one ACL6 and apply all of them at the same time.

Note: ACL6s created on the NetScaler do not work until they are applied.

To apply ACL6s by using the NetScaler command line

At the NetScaler command prompt, type:

apply ns acls6

To apply ACL6s by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. Click **Commit**.
3. In the **Apply ACL(s)** dialog box that appears, click **Yes**.
4. Verify that the settings displayed on the **ACL6s** tab are correct.

Enabling and Disabling ACL6s

By default, ACL6s are enabled. Therefore, after the ACL6s are applied, the NetScaler appliance compares incoming packets against the configured ACL6s.

If an ACL6 is not required to be part of the lookup table but needs to be retained in the configuration, it must be disabled before the ACL6s are applied. After the ACL6s are applied, the NetScaler does not compare incoming packets against disabled ACL6s.

To disable or enable an ACL6 by using the NetScaler command line

At the NetScaler command prompt, type:

- enable ns acl6 <acl6name>
- show ns acl6 [<acl6name>]
- disable ns acl6 <acl6name>
- show ns acl6 [<acl6name>]

Note: ACL6s created on the NetScaler do not work until they are applied.

Example

```
> enable ns acl6 rule6
 Done

> show ns acl6 rule6
      Name: rule6                    Action: DENY
      srcIPv6
      destIPv6 = 2001::45
      srcMac:                      Protocol: TCP
      srcPort =  45-1024               destPort
      Vlan:                       Interface:
      Active Status: ENABLED             Applied Status: NOTAPPLIED
      Priority: 10                 Hits: 0
      TTL:
 Done

> disable ns acl6 rule6
 Done
```

```
> show ns acl6 rule6
      Name: rule6                    Action: DENY
      srcIPv6
      destIPv6 = 2001::45
      srcMac:                      Protocol: TCP
      srcPort =  45-1024              destPort
      Vlan:                       Interface:
      Active Status: DISABLED           Applied Status: NOTAPPLIED
      Priority: 10                 Hits: 0
      TTL:
Done
```

To disable or enable an ACL6 by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **ACL6s** tab, select the ACL (for example, rule1) and do one of the following:
   - To disable the extended ACL6, click **Disable**.
   - To enable the extended ACL6, click **Enable**.
3. If you want to apply the new setting, which reapplies all ACLs, click **Commit**, and then, in the **Apply ACL6(s)** dialog box, click **Yes**.
4. In the details pane, on the **Extended ACL6s** tab, view the list to verify the changed status under the column **Active Status**.

Renumbering the Priority of ACL6s

The renumber procedure resets the priorities of the ACL6s to multiples of 10. The priority (an integer value) defines the order in which the NetScaler appliance evaluates ACL6s. All priorities are multiples of 10, unless you configure a specific priority to an integer value. When you create an ACL6 without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the ACL6, the NetScaler performs an action. If the packet does not match the condition defined by the ACL6, the NetScaler compares the packet against the ACL6 with the next-highest priority.

Consider the following example. Two ACL6s, rule1 and rule2, are automatically assigned priorities 20 and 30 when they are created. You need to add a third ACL, rule3, to be evaluated immediately after rule1. Rule3 must have a priority between 20 and 30. In this case, you can specify the priority as 25. Later, you can easily renumber the ACL6s with priorities that are multiples of 10, without affecting the order in which the ACLs6are applied.

To renumber the priorities of the ACL6s by using the NetScaler command line

At the NetScaler command prompt, type:

renumber ns acls6

Example

```
> renumber ns acls6
Done
```

To renumber the priority of ACL6s by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the ACL6s tab, click **Renumber Priority (s)** ACL(s).
3. In the **Renumber Priority (s)** ACL(s) dialog box, click **Yes**.
4. Verify the action in the details pane.

Monitoring ACL6s

You can display statistics for monitoring the performance of an ACL6.

To display the statistics for an ACL6s by using the NetScaler command line

At the NetScaler command prompt, type:

stat ns acl6 <acl6name>

Example

```
> stat ns acl6 rule6
ACL6: rule6
                       Rate (/s)          Total
Hits for this ACL6             0              0
 Done
```

The following table lists the statistics associated with ACL6s and their descriptions.

Table 1. ACL6 Statistics

Statistic	Specifies
Allow ACL6 hits	Packets matching IPv6 ACLs with processing mode set to ALLOW. The NetScaler processes these packets.
NAT ACL6 hits	Packets matching a NAT ACL6, resulting in a NAT session.
Deny ACL6 hits	Packets dropped because they match IPv6 ACLs with processing mode set to DENY.
Bridge ACL6 hits	Packets matching a bridge IPv6 ACL, which in transparent mode bypasses service processing.
ACL6 hits	Packets matching an IPv6 ACL.
ACL6 misses	Packets not matching any IPv6 ACL.

To display the statistics for an ACL6 by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **ACL6s** tab, select the ACL whose statistics you want to view (for example, rule1).
3. Click **Statistics**.
4. View the statistics in the new window that opens.

Removing ACL6s

You can remove a single ACL6 or all ACL6s.

To remove an extended ACL6 by using the NetScaler command line

At the NetScaler command prompt, type:

- rm ns acl6 <acl6name>
- show ns acl6

To remove all extended ACL6s by using the NetScaler command line

At the NetScaler command prompt, type:

clear ns acls6

To remove an extended ACL6 by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **ACL6s** tab, select the ACL that you want to remove (for example, rule1).
3. Click **Remove**.
4. In the **Remove** dialog box, click **Yes**.
5. In the details pane, on the **ACL6s** tab, verify that the ACL6 is not listed

To remove all extended ACLs by using the configuration utility

1. In the navigation pane, expand **Network** and click **ACLs**.
2. In the details pane, on the **ACL6s** tab, click **Clear**.
3. In the **Clear ACL (s)** dialog box, click **Yes**
4. In the details pane, on the **Extended ACLs** tab, verify that no ACLs are listed

IP Routing

NetScaler® appliances support both dynamic and static routing. Because simple routing is not the primary role of a NetScaler, the main objective of running dynamic routing protocols is to enable route health injection (RHI), so that an upstream router can choose the best among multiple routes to a topographically distributed virtual server.

Most NetScaler implementations use some static routes to reduce routing overhead. You can create backup static routes and monitor routes to enable automatic switchover in the event that a static route goes down. You can also assign weights to facilitate load balancing among static routes, create null routes to prevent routing loops, and configure IPv6 static routes. You can configure policy based routes (PBRs), for which routing decisions are based on criteria that you specify

Configuring Dynamic Routes

When a dynamic routing protocol is enabled, the corresponding routing process monitors route updates and advertises routes. Routing protocols enable an upstream router to use the equal cost multipath (ECMP) technique to load balance traffic to identical virtual servers hosted on two standalone NetScaler appliances. Dynamic routing on a NetScaler appliance uses three routing tables. In a high-availability setup, the routing tables on the secondary appliance mirror those on the primary.

The NetScaler supports the following protocols:

- Routing Information Protocol (RIP) version 2
- Open Shortest Path First (OSPF) version 2
- Border Gateway Protocol (BGP)
- Routing Information Protocol next generation (RIPng) for IPv6
- Open Shortest Path First (OSPF) version 3 for IPv6

You can enable more than one protocol simultaneously.

Routing Tables in the NetScaler

In a NetScaler appliance, the NetScaler kernel routing table, the FreeBSD kernel routing table, and the NSM FIB routing table each hold a different set of routes and serve a different purpose. They communicate with each other by using UNIX routing sockets. Route updates are not automatically propagated from one routing table to another. You must configure propagation of route updates for each routing table.

NS Kernel Routing Table

The NS kernel routing table holds subnet routes corresponding to the NSIP and to each SNIP and MIP. Usually, no routes corresponding to VIPs are present in the NS kernel routing table. The exception is a VIP added by using the `add ns ip` command and configured with a subnet mask other than 255.255.255.255. If there are multiple IP addresses belonging to the same subnet, they are abstracted as a single subnet route. In addition, this table holds a route to the loopback network (127.0.0.0) and any static routes added through the NetScaler command-line interface (CLI). The entries in this table are used by the NetScaler in packet forwarding. From the NetScaler CLI, they can be inspected with the `show route` command.

FreeBSD Routing Table

The sole purpose of the FreeBSD routing table is to facilitate initiation and termination of management traffic (telnet, ssh, etc.). In a NetScaler appliance, these applications are tightly coupled to FreeBSD, and it is imperative for FreeBSD to have the necessary information to handle traffic to and from these applications. This routing table contains a route to the NSIP subnet and a default route. In addition, FreeBSD adds routes of type `WasCloned(W)` when the NetScaler establishes connections to hosts on local networks. Because of the highly specialized utility of the entries in this routing table, all other route updates from the NS kernel and NSM FIB routing tables bypass the FreeBSD routing table. Do not modify it with the `route` command. The FreeBSD routing table can be inspected by using the `netstat` command from any UNIX shell.

Network Services Module (NSM) FIB

The NSM FIB routing table contains the advertisable routes that are distributed by the dynamic routing protocols to their peers in the network. It may contain:

Connected routes

IP subnets that are directly reachable from the NetScaler. Typically, routes corresponding to the NSIP subnet and subnets over which routing protocols are enabled are present in NSM FIB as connected routes.

Kernel routes

All the VIP addresses on which the **-hostRoute** option is enabled are present in NSM FIB as kernel routes if they satisfy the required RHI Levels. In addition, NSM FIB contains any static routes configured on the NetScaler CLI that have the - advertise option enabled. Alternatively, if the NetScaler is operating in Static Route Advertisement (SRADV) mode, all static routes configured on the NetScaler CLI are present in NSM FIB. These static routes are marked as kernel routes in NSM FIB, because they actually belong to the NS kernel routing table.

Static routes

Normally, any static route configured in VTYSH is present in NSM FIB. If administrative distances of protocols are modified, this may not always be the case. An important point to note is that these routes can never get into the NS kernel routing table.

Learned routes

If the NetScaler is configured to learn routes dynamically, the NSM FIB contains routes learned by the various dynamic routing protocols. Routes learned by OSPF, however, need special processing. They are downloaded to FIB only if the **fib-install** option is enabled for the OSPF process. This can be done from the router-config view in VTYSH.

High Availability Setup

In a high availability setup, the primary node runs the routing process and propagates routing table updates to the secondary node. The routing table of the secondary node mirrors the routing table on the primary node.

Non-Stop Forwarding

After failover, the secondary node takes some time to start the protocol, learn the routes, and update its routing table. But this does not affect routing, because the routing table on the secondary node is identical to the routing table on the primary node. This mode of operation is known as non-stop forwarding.

Black Hole Avoidance Mechanism

After failover, the new primary node injects all its VIP routes into the upstream router. However, that router retains the old primary node's routes for 180 seconds. Because the router is not aware of the failover, it attempts to load balance traffic between the two nodes. During the 180 seconds before the old routes expire, the router sends half the traffic to the old, inactive primary node, which is, in effect, a black hole.

To prevent this, the new primary node, when injecting a route, assigns it a metric that is slightly lower than the one specified by the old primary node.

Interfaces for Configuring Dynamic Routing

To configure dynamic routing, you can use either the configuration utility or a command-line interface. The NetScaler supports two independent command-line interfaces: the NetScaler CLI and the Virtual Teletype Shell (VTYSH). The NetScaler CLI is the appliance's native shell. VTYSH is exposed by ZebOS. The NetScaler routing suite is based on ZebOS, the commercial version of GNU Zebra.

Note: Citrix recommends that you use VTYSH for all commands except those that can be configured only on the NetScaler CLI. Use of the NetScaler CLI should generally be limited to commands for enabling the routing protocols, configuring host route advertisement, and adding static routes for packet forwarding.

Configuring RIP

Routing Information Protocol (RIP) is a Distance Vector protocol. The NetScaler supports RIP as defined in RFC 1058 and RFC 2453. RIP can run on any subnet.

After enabling RIP, you need to configure advertisement of RIP routes. For troubleshooting, you can limit RIP propagation. You can display RIP settings to verify the configuration.

Enabling and Disabling RIP

Use either of the following procedures to enable or disable RIP. After you enable RIP, the NetScaler appliance starts the RIP process. After you disable RIP, the appliance stops the RIP process.

To enable or disable RIP routing by using the NetScaler command line

At the NetScaler command prompt, enter one of the following commands to enable or disable RIP:

- enable ns feature RIP
- disable ns feature RIP

To enable or disable RIP routing by using the configuration utility

1. In the navigation pane, expand **System,** and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, do one of the following:
   - To enable OSPF routing, select the **RIP Routing** check box.
   - To disable OSPF routing, clear the **RIP Routing** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Advertising Routes

RIP enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertisement enables an upstream router to track network entities located behind the NetScaler.

To configure RIP to advertise routes by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting RIP Propagations

If you need to troubleshoot your configuration, you can configure listen-only mode on any given interface.

To limit RIP propagation by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
```

Verifying the RIP Configuration

You can display the routing table and other RIP settings.

To view the RIP settings by using the VTYSH command line

At the NetScaler command prompt, type the following commands in the following order:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh rip	Display updated RIP routing table.
sh rip interface <vlan_name>	Displays RIP information for the specified VLAN.

Example

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
```

Configuring OSPF

The NetScaler supports Open Shortest Path First (OSPF) Version 2 (RFC 2328). The features of OSPF on the NetScaler are:

- The NetScaler supports OSPF within a single area only.
- If a vserver is active, the host routes to the vserver can be injected into the routing protocols.
- OSPF can run on any subnet.
- Route learning advertised by neighboring OSPF routers can be disabled on the NetScaler.
- The NetScaler can advertise Type-1 or Type-2 external metrics for all routes.
- The NetScaler can advertise user-specified metric settings for VIP routes. For example, you can configure a metric per VIP without special route maps.
- You can specify the OSPF area ID for the NetScaler.
- The NetScaler supports not-so-stubby-areas (NSSAs). An NSSA is similar to an OSPF stub area but allows injection of external routes in a limited fashion into the stub area. To support NSSAs, a new option bit (the N bit) and a new type (Type 7) of Link State Advertisement (LSA) area have been defined. Type 7 LSAs support external route information within an NSSA. An NSSA area border router (ABR) translates a type 7 LSA into a type 5 LSA that is propagated into the OSPF domain. The OSPF specification defines only the following general classes of area configuration:
    - Type 5 LSA: Originated by routers internal to the area are flooded into the domain by AS boarder routers (ASBRs).
    - Stub: Allows no type 5 LSAs to be propagated into/throughout the area and instead depends on default routing to external destinations.

After enabling OSPF, you need to configure advertisement of OSPF routes. For troubleshooting, you can limit OSPF propagation. You can display OSPF settings to verify the configuration.

Enabling and Disabling OSPF

To enable or disable OSPF, you must use either the NetScaler command line or the configuration utility. When OSPF is enabled, the NetScaler starts the OSPF process. When OSPF is disabled, the NetScaler stops the OSPF routing process.

To enable or disable OSPF routing by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

1. enable ns feature OSPF
2. disable ns feature OSPF

To enable or disable OSPF routing by using the configuration utility

1. In the navigation pane, expand **System,** and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, do one of the following:
   - To enable OSPF routing, select the **OSPF Routing** check box.
   - To disable OSPF routing, clear the **OSPF Routing** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Advertising OSPF Routes

OSPF enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure OSPF to advertise routes by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enters global configuration mode.
router OSPF	Start OSPF routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting OSPF Propagations

If you need to troubleshoot your configuration, you can configure listen-only mode on any given VLAN.

To limit OSPF propagation by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router OSPF	Start OSPF routing process and enters configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
```

Verifying the OSPF Configuration

You can display current OSPF neighbors, and OSPF routes.

To view the OSPF settings by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh OSPF neighbor	Displays current neighbors.
sh OSPF route	Displays OSPF routes.

Example

```
>VTYSH
NS# sh OSPF neighbor
NS# sh OSPF route
```

Configuring BGP

The NetScaler appliance supports BGP (RFC 4271). The features of BGP on the NetScaler are:

- The NetScaler advertises routes to BGP peers.
- The NetScaler injects host routes to virtual IP addresses (VIPs), as determined by the health of the underlying virtual servers.
- The NetScaler generates configuration files for running BGP on the secondary node after failover in an HA configuration.
- This protocol supports IPv6 route exchanges.

After enabling BGP, you need to configure advertisement of BGP routes. For troubleshooting, you can limit BGP propagation. You can display BGP settings to verify the configuration.

Prerequisites for IPv6 BGP

Before you begin configuring IPv6 BGP, do the following:

- Make sure that you understand the IPv6 BGP protocol.
- Install the IPv6PT license on the NetScaler appliance.
- After installing the IPv6PT license, enable the IPv6 feature.

Enabling and Disabling BGP

To enable or disable BGP, you must use either the NetScaler command line or the configuration utility. When BGP is enabled, the NetScaler appliance starts the BGP process. When BGP is disabled, the appliance stops the BGP process.

To enable or disable BGP routing by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable ns feature BGP
- disable ns feature BGP

To enable or disable BGP routing by using the configuration utility

1. In the navigation pane, expand **System** and click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, do one of the following:
   - To enable BGP routing, select the **BGP Routing** check box.
   - To disable BGP routing, clear the **BGP Routing** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Advertising IPv4 Routes

You can configure the NetScaler appliance to advertise host routes to VIPs and to advertise routes to downstream networks.

To configure BGP to advertise IPv4 routes by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < Asnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv4 address> **remote-as** < as-number>	Update the IPv4 BGP neighbor table with the link local IPv4 address of the neighbor in the specified autonomous system.
Address-family ipv4	Enter address family configuration mode.
Neighbor < IPv4 address> **activate**	Exchange prefixes for the IPv4 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv4
NS(config-router-af)# Neighbor 10.102.29.170 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
```

Advertising IPv6 BGP Routes

Border Gateway Protocol (BGP) enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure BGP to advertise IPv6 routes by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < Asnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv6 address> **remote-as** < as-number>	Update the IPv6 BGP neighbor table with the link local IPv6 address of the neighbor in the specified autonomous system.
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> **activate**	Exchange prefixes for the IPv6 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
```

Verifying the BGP Configuration

You can use VTYSH to display BGP settings.

To view the BGP settings using the VTYSH command line

At the NetScaler command prompt, type:

VTYSH
You are now in the VTYSH command prompt. An output similar to the following appears:
NS170#
At the VTYSH command prompt, type:
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>

Configuring IPv6 RIP

IPv6 Routing Information Protocol (RIP) or RIPng is a Distance Vector protocol. This protocol is an extension of RIP to support IPv6. After enabling IPv6 RIP, you need to configure advertisement of IPv6 RIP routes. For troubleshooting, you can limit IPv6 RIP propagation. You can display IPv6 RIP settings to verify the configuration.

Prerequisites for IPv6 RIP

Before you begin configuring IPv6 RIP, do the following:

- Make sure that you understand the IPv6 RIP protocol.
- Install the IPv6PT license on the NetScaler appliance.
- Enable the IPv6 feature.

Enabling IPv6 RIP

You can enable or disable IPv6 RIP by using VTYSH. After you enable IPv6 RIP, the NetScaler starts the IPv6 RIP daemon. After you disable IPv6 RIP, the NetScaler stops the RIP daemon.

To enable IPv6 RIP by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns IPv6-routing	Start IPv6 dynamic routing daemon.
interface < vlan_name>	Enter VLAN configuration mode.
router ipv6 RIP	Start IPv6 RIP routing process on the VLAN.

Example

```
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)# router ipv6 RIP
```

Advertising IPv6 RIP Routes

IPv6 RIP enables an upstream router to load balance traffic between two identical vservers hosted on two standalone NetScaler devices. Route advertisement enables an upstream router to track network entities located behind the NetScaler.

To configure IPv6 RIP to advertise IPv6 routes by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting IPv6 RIP Propagations

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given interface.

To limit IPv6 RIP propagation by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# passive-interface VLAN0
```

Verifying the IPv6 RIP Configuration

You can use VTYSH to display the IPv6 RIP routing table and IPv6 RIP information for a specified VLAN.

To view the IPv6 RIP settings by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
sh ipv6 rip	Display updated IPv6 RIP routing table.
sh ipv6 rip interface <vlan_name>	Display IPv6 RIP information for the specified VLAN.

Example

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
```

Configuring IPv6 OSPF

IPv6 OSPF or OSPF version 3 (OSPF v3) is a link state protocol that is used to exchange IPv6 routing information. After enabling IPv6 OSPF, you need to configure advertisement of IPv6 OSPF routes. For troubleshooting, you can limit IPv6 OSPF propagation. You can display IPv6 OSPF settings to verify the configuration.

Prerequisites for IPv6 OSPF

Before you begin configuring IPv6 OSPF, do the following:

- Make sure that you understand the IPv6 OSPF protocol.
- Install the IPv6PT license on the NetScaler appliance.
- Enable the IPv6 feature.

Enabling IPv6 OSPF

To enable IPv6 OSPF, you must use the VTYSH command line. When IPv6 OSPF is enabled, the NetScaler appliance starts the IPv6 OSPF daemon. When IPv6 OSPF is disabled, the appliance stops the IPv6 OSPF daemon.

To enable IPv6 OSPF by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns IPv6-routing	Start IPv6 dynamic routing process.
interface < vlan_name>	Enter the VLAN configuration mode.
ipv6 router OSPF area < area-id>	Start IPv6 OSPF routing process on a VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)# ipv6 router OSPF area 3
```

Advertising IPv6 Routes

IPv6 OSPF enables an upstream router to load balance traffic between two identical vservers hosted on two standalone NetScaler devices. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure IPv6 OSPF to advertise IPv6 routes by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting IPv6 OSPF Propagations

If you need to troubleshoot your configuration, you use VTYSH to configure listen-only mode on any given VLAN.

To limit IPv6 OSPF propagation by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
passive-interface < vlan_name >	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# passive-interface VLAN0
```

Verifying the IPv6 OSPF Configuration

You use VTYSH to display IPv6 OSPF current neighbors and IPv6 OSPF routes.

To view the IPv6 OSPF settings by using the VTYSH command line

At the NetScaler command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh ipv6 OSPF neighbor	Display current neighbors.
sh ipv6 OSPF route	Display IPv6 OSPF routes.

Example

```
>VTYSH
NS# sh ipv6 OSPF neighbor
NS# sh ipv6 OSPF route
```

Installing Routes to the NetScaler Routing Table

The NetScaler appliance can use routes learned by various routing protocols after you install the routes in the appliance's routing table.

To install various routes to the internal routing table by using the VTYSH command line

At the NetScaler command prompt, type the following commands as appropriate for the routes that you want to install:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns route-install Default	Install IPv4 default routes to the internal routing table.
ns route-install RIP	Install IPv4 RIP specific routes to the internal routing table.
ns route-install BGP	Install IPv4 BGP specific routes to the internal routing table.
ns route-install OSPF	Install IPv4 OSPF specific routes to the internal routing table.
ns route-install IPv6 Default	Install IPv6 default routes to the internal routing table.
ns route-install IPv6 RIP	Install IPv6 RIP specific routes to the internal routing table.
ns route-install IPv6 BGP	Install IPv6 BGP specific routes to the internal routing table.
ns route-install IPv6 OSPF	Install IPv6 OSPF specific routes to the internal routing table.

Example

```
>VTYSH
NS# configure terminal
NS# ns route-install Default
```

```
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
NS(config)# ns route-install IPv6 BGP
NS(config)# ns route-install IPv6 OSPF
```

Configuring Static Routes

Static routes are manually created to improve the performance of your network. You can monitor static routes to avoid service disruptions. Also, you can assign weights to ECMP routes, and you can create null routes to prevent routing loops.

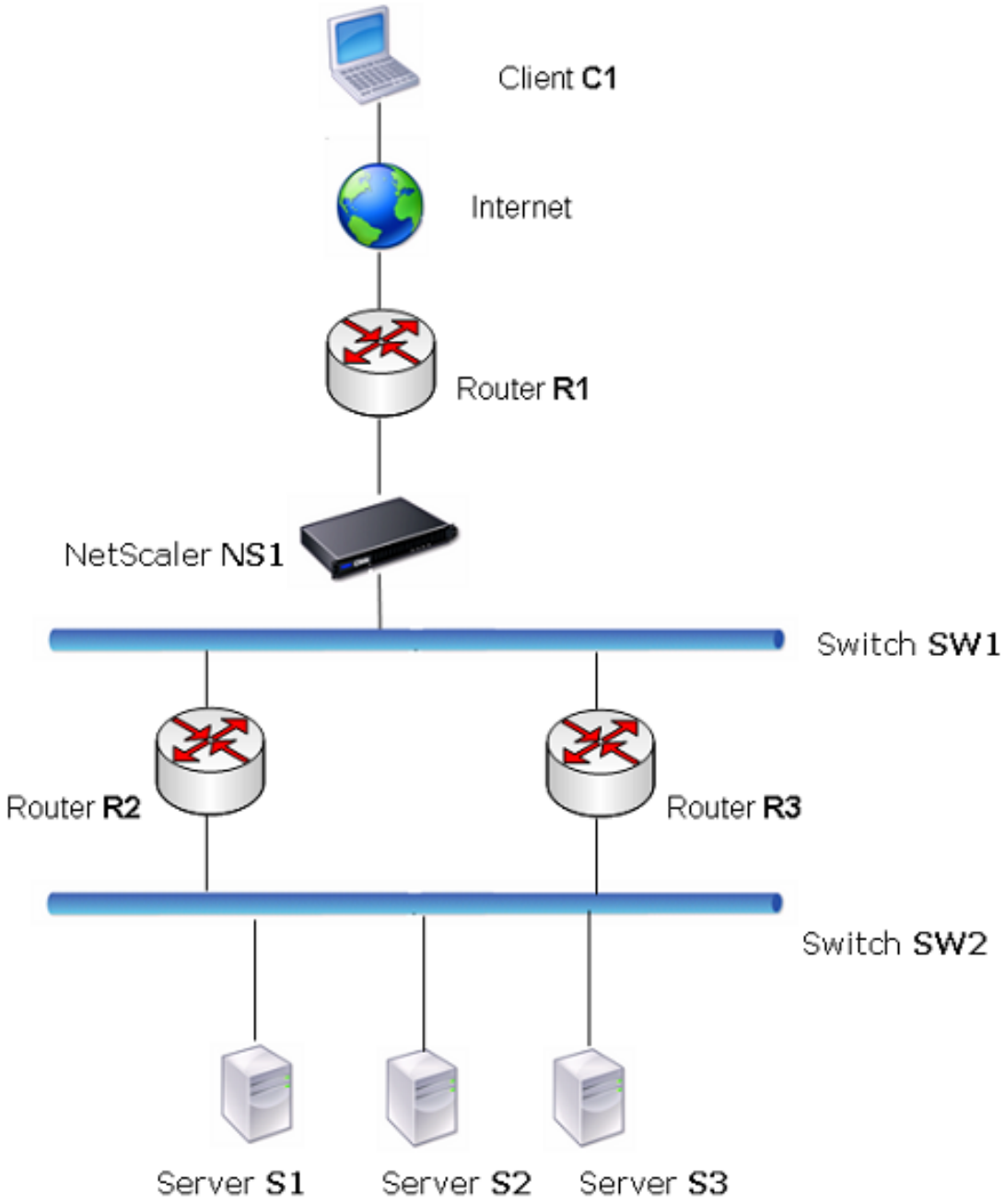
Monitored Static Routes

If a manually created (static) route goes down, a backup route is not automatically activated. You must manually delete the inactive primary static route. However, if you configure the static route as a monitored route, the NetScaler appliance can automatically activate a backup route.

Static route monitoring can also be based on the accessibility of the subnet. A subnet is usually connected to a single interface, but it can be logically accessed through other interfaces. Subnets bound to a VLAN are accessible only if the VLAN is up. VLANs are logical interfaces through which packets are transmitted and received by the NetScaler. A static route is marked as DOWN if the next hop resides on a subnet that is unreachable.

Note: In a high availability (HA) setup, the default value for monitored state routes (MSRs) on the secondary node is UP. The value is set to avoid a state transition gap upon failover, which could result in dropping packets on those routes.

Consider the following simple topology, in which a NetScaler is load balancing traffic to a site across multiple servers.



Router R1 moves traffic between the client and the NetScaler appliance. The appliance can reach servers S1 and S2 through routers R2 or R3. It has two static routes through which to reach the servers' subnet, one with R2 as the gateway and another with R3 as the gateway. Both these routes have monitoring enabled. The administrative distance of the static route with gateway R2 is lower than that of the static route with gateway R3. Therefore, R2 is preferred over R3 to forward traffic to the servers. Also, the default route on the NetScaler points to R1 so that all Internet traffic exits properly.

If R2 fails while monitoring is enabled on the static route, which uses R2 as the gateway, the NetScaler marks it as DOWN. The NetScaler now uses the static route with R3 as the gateway and forwards the traffic to the servers through R3.

The NetScaler supports monitoring of IPv4 and IPv6 static routes. You can configure the NetScaler to monitor an IPv4 static route either by creating a new ARP or PING monitor or by using existing ARP or PING monitors. You can configure the NetScaler to monitor an IPv6 static route either by creating a new Neighbor discovery for IPv6 (ND6) or PING monitor or by using the existing ND6 or PING monitors.

Weighted Static Routes

When the NetScaler appliance makes routing decisions involving routes with equal distance and cost, that is, Equal Cost Multi-Path (ECMP) routes, it balances the load between them by using a hashing mechanism based on the source and destination IP addresses. For an ECMP route, however, you can configure a weight value. The NetScaler then uses both the weight and the hashed value for balancing the load.

Null Routes

If the route chosen in a routing decision is inactive, the NetScaler appliance chooses a backup route. If all the backup routes become inaccessible, the appliance might reroute the packet to the sender, which could result in a routing loop leading to network congestion. To prevent this situation, you can create a null route, which adds a null interface as a gateway. The null route is never the preferred route, because it has a higher administrative distance than the other static routes. But it is selected if the other static routes become inaccessible. In that case, the appliance drops the packet and prevents a routing loop.

Configuring IPv4 Static Routes

You can add a simple static route or a null route by setting a few parameters, or you can set additional parameters to configure a monitored or monitored and weighted static route. You can change the parameters of a static route. For example, you might want to assign a weight to an unweighted route, or you might want to disable monitoring on a monitored route.

To create a static route by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a static route and verify the configuration:

- add route <network> <netmask> <gateway>[**-cost** <positive_integer>] [**-advertise** (DISABLED | ENABLED)]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [**-detail**]

Example

```
> add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise ENABLED
 Done
> show route 10.102.29.0 255.255.255.0 10.102.29.2
      Network         Netmask         Gateway/OwnedIP      State   Type
      -------         -------         ---------------      -----   ----
1)     10.102.29.0      255.255.255.0    10.102.29.2         UP      STATIC
      Distance:     1 Cost:     2     Weight:     1

 Done
```

To create a monitored static route by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a monitored static route and verify the configuration:

- add route <network> <netmask> <gateway> [**-distance** <positive_integer>] [**-weight** <positive_integer>][**-msr** (ENABLED | DISABLED) [**-monitor** <string>]]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [**-detail**]

Example

```
> add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6 -msr ENBLED -monitor PING
 Done
> show route 10.102.29.0 255.255.255.0 10.102.29.3
      Network         Netmask         Gateway/OwnedIP     State   Type
      -------         -------         ---------------     -----   ----
1)     10.102.29.0     255.255.255.0   10.102.29.3         UP    STATIC
      Distance:     5 Cost:     0     Weight:     6
      MSR: ENABLED    Monitor: ping
          Probes:         3 Failed: [Total: 3 Current: 3]
          Last response: Failure - Probe timed out.
Done
```

To create a null route by using the NetScaler command line

At the NetScaler command prompt type:

- add route <network> <netmask> **null**
- show route <network> <netmask>

Example

```
> add route 10.102.29.0 255.255.255.0 null
 Done
> show route 10.102.29.0 255.255.255.0
      Network         Netmask         Gateway/OwnedIP     State   Type
      -------         -------         ---------------     -----   ----
1)     10.102.29.0     255.255.255.0   10.102.29.200       UP      DIRECT
2)     10.102.29.0     255.255.255.0   null                UP      STATIC
3)     10.102.29.0     255.255.255.0   10.102.29.1         UP      STATIC|ADV
4)     10.102.29.0     255.255.255.0   10.102.29.2         UP      STATIC
5)     10.102.29.0     255.255.255.0   10.102.29.3         DOWN    STATIC
 Done
```

To remove a static route by using the NetScaler command line

At the NetScaler command prompt, type:

rm route <network> <netmask> <gateway>

Example

```
> rm route 10.102.29.0 255.255.255.0 10.102.29.3
 Done
```

Parameters for configuring static routes

network

Network for which the route is being created.

netmask

Subnet mask for the network

null

Drop the packets this route receives. Possible values: Yes, No. Default: No. Null routes have a fixed distance of 255.

gateway

IP address of the gateway for this route.

distance

Administrative distance of this route. Possible values: 1 through 255. Default: 1.

cost

Value used by the routing algorithms to compare performance. Route having lowest cost is the most preferred route. Value that this parameter can take is from 0 through 65535.

weight

Value to facilitate balancing the load on ECMP routes. This value is compared with the hashed value of the packet and a route is chosen. Specific to ECMP routes. Possible values: 1 through 65535. Default: 1.

advertise

State of advertisement of this route. Possible values: Enabled or Disabled. Default: Enabled.

protocol

Routing protocols used for advertising routes. Possible values: OSPF, RIP, BGP.

msr

Monitor this route. Possible values: Enabled, Disabled. Default: Disabled.

monitor

Type of monitor. Determines the protocol used for monitoring the route (for example, PING or ARP).

To configure a static route by using the configuration utility

1. In the navigation pane, expand **Network**, expand **Routing**, and then click **Routes**.
2. In the details pane, on the **Basic** tab, do one of the following:
   - To create a new static route, click **Add**.
   - To modify an existing static route, click **Open**.
3. In the **Create Route** or **Configure Route** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring static routes” as shown:
   - **Network***-network
   - **Netmask***-netmask
   - **Null Route**-null
   - **Gateway***-gateway
   - **Distance**-distance
   - **Cost**-cost
   - **Weight**-weight
   - **Over-ride Global**-advertise
   - **Protocol**-protocol
   - **Monitored Static Route**-msr
   - **Monitor**-monitor

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the route has been configured successfully.

To remove a route by using the configuration utility

1. In the navigation pane, expand **Network**, expand **Routing**, and then click **Routes**.
2. On the **Routes** pane, click the **Basic** tab, select the route you want to remove (for example, **192.168. 20.2**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**. A message appears in the status bar, stating that the route has been successfully removed.

Configuring IPv6 Static Routes

You can configure a maximum of six default IPv6 static routes. IPv6 routes are selected on the basis of whether the MAC address of the destination device is reachable. This can be determined by using the IPv6 Neighbor Discovery feature. Routes are load balanced and only source/destination-based hash mechanisms are used. Therefore, route selection mechanisms such as round robin are not supported. The next hop address in the default route need not belong to the NSIP subnet.

To create an IPv6 route by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an IPv6 route and verify the configuration:

- add route6 <network> <gateway> [**-vlan** <positive_integer>]
- show route6 [<network> [<gateway>]

Example

```
> add route6 ::/0 FE80::67 -vlan 5
 Done

> show route6
Flags: S - Static, C - Connected, R - RA Route, A - Active, O - OSPFV3, P – Permanent

Network                 Gateway                 Vlan      Flags
-------                 -------                 ----      -----

::1/128                   ::1                     1        PA
::/0                    fe80::67                  5        SA
fe80::/64                 fe80::2d0:68ff:fe15:fd36   1        CA

 Done
```

To create a monitored IPv6 static route by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a monitored IPv6 static route and verify the configuration:

- add route6 <network> <gateway> [**-msr** (ENABLED | DISABLED) [**-monitor** <string>]]
- show route6 [<network> [<gateway>]

Example

```
> add route6 ::/0 2004::1 -msr ENABLED -monitor PING
 Done
> show route6
Flags: S - Static, C - Connected, R - RA Route, A - Active, O - OSPFV3, n - RIPng, B - BGP, P - Permanent
Network                 Gateway                 Vlan      Flags
-------                 -------                 ----      -----
::1/128                   ::1                     1        PA
fe80::/64                 fe80::2d0:68ff:fe17:33c  1        CA
::/0                    2004::1                   0        S
 Done
```

To remove an IPv6 route by using the NetScaler command line

At the NetScaler command prompt, type:

rm route6 <network> <gateway>

Example

```
> rm route6 ::/0 FE80::67
 Done
```

Parameters for configuring IPv6 static routes

network

Network for which the route is being created.

gateway

IP address of the gateway for this route.

vlan

Virtual LAN (VLAN) number associated with the route. Possible values: 1 through 4094. Default: 0. Required for link-local address type.

distance

Administrative distance of this route. Possible values: 1 through 255. Default: 1.

cost

Value used by the routing algorithms to compare performance. Route having lowest cost is the most preferred route. Possible values: 0 through 65535.

weight

Value for balancing the load on ECMP routes. This value is compared with the hashed value of the packet and a route is chosen. Specific to ECMP routes. Possible values: 1 through 65535. Default: 1.

advertise

Advertise this route. Possible values: Enabled, Disabled. Default: Enabled.

msr

Monitor this route. Possible values: Enabled, Disabled. Default: Disabled.

monitor

A ND6 or a PING monitor that will be used for monitoring the IPv6 static route.

To configure an IPv6 route by using the configuration utility

1. In the navigation pane, expand **Network**, expand **Routing**, and then click **Routes**.
2. In the details pane, on the **IPv6** tab, do one of the following:
   - To create a new route, click **Add**.
   - To modify an existing route, click **Open**.
3. In the **Create IPv6 Route** or **Configure IPv6 Route** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring IPv6 static routes” as shown:
   - **Network***-network
   - **Gateway***-gateway
   - **VLAN**-vlan
   - **Distance**-distance
   - **Cost**-cost
   - **Weight**-weight
   - **Advertise**-advertise
   - **Monitored Static Route**-msr
   - **Monitor**-monitor

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the IPv6 route has been configured successfully.

To remove an IPv6 route by using the configuration utility

1. In the navigation pane, expand **Network**, expand **Routing**, and then click **Routes**.
2. On the **Routes** pane, click the **IPV6** tab.
3. Select the network, from which you want to remove the route (for example,**::/0**), and then click **Remove**.
4. In the **Remove** dialog box, click **Yes**. A message appears in the status bar, stating that the IPv6 route has been successfully removed.

Configuring Policy-Based Routes

Policy-based routing bases routing decisions on criteria that you specify. A policy-based route (PBR) specifies criteria for selecting packets and, typically, a next hop to which to send the selected packets. For example, you can configure the NetScaler appliance to route outgoing packets from a specific IP address or range to a particular next hop router. Each packet is matched against each configured PBR, in the order determined by the specified priorities, until a match is found. If no match is found, or if the matching PBR specifies a DENY action, the NetScaler applies the routing table for normal destination-based routing.

Instead of sending the selected packets to a next hop router, you can send them to a link load balancing virtual server to which you have bound multiple next hops. This configuration can provide a backup if a next hop link fails.

A PBR bases routing decisions for the data packets on parameters such as source IP address, source port, destination IP address, destination port, protocol, and source MAC address. A PBR defines the conditions that a packet must satisfy for the NetScaler to route the packet. These actions are known as "processing modes." The processing modes are:

- ALLOW - The NetScaler sends the packet to the designated next-hop router.
- DENY - The NetScaler applies the routing table for normal destination-based routing.

The NetScaler process PBRs before processing the RNAT rules.

Many users begin by creating PBRs and then modifying them. To activate a new PBR, you must apply it. To deactivate a PBR, you can either remove or disable it. You can change the priority number of a PBR to give it a higher or lower precedence.

Creating or Modifying a PBR

You cannot create two PBRs with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR. The priority (an integer value) defines the order in which the NetScaler appliance evaluates PBRs. When you create a PBR without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR, the NetScaler performs an action. If the packet does not match the condition defined by the PBR, the NetScaler compares the packet against the PBR with the next highest priority.

Consider the following example. Two PBRs, p1 and p2, are configured on the NetScaler and automatically assigned priorities 20 and 30. You need to add a third PBR, p3, to be evaluated immediately after the first PBR, p1. The new PBR, p3, must have a priority between 20 and 30. In this case, you can specify the priority as 25.

To create a PBR by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns pbr <name> <action> [**-srcIP** [<operator>] <srcIPVal>] [**-srcPort** [<operator>] <srcPortVal>] [**-destIP** [<operator>] <destIPVal>] [**-destPort** [<operator>] <destPortVal>] [**-nextHop** <nextHopVal>] [**-srcMac** <mac_addr>] [**-protocol** <protocol> |**-protocolNumber** <positive_integer>] [**-vlan** <positive_integer>] [**-interface** <interface_name>] [**-priority** <positive_integer>] [**-msr** (**ENABLED** | **DISABLED**) [**-monitor** <string>]] [**-state** (**ENABLED** | **DISABLED**)]

- sh ns pbr

Example

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -nexthop 10.102.29.77
 Done
> sh ns pbr pbr1
1)      Name: pbr1
        Action: ALLOW                   Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                      Protocol:
        Vlan:                       Interface:
        Active Status: ENABLED            Applied Status: NOTAPPLIED
        Priority: 10
        NextHop: 10.102.29.77
```

Done

To modify the priority of a PBR by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify the priority and verify the configuration:

- set ns pbr <name> [**-action** (ALLOW | DENY)] [**-srcIP** [<operator>] <srcIPVal>] [**-srcPort** [<operator>] <srcPortVal>] [**-destIP** [<operator>] <destIPVal>] [**-destPort** [<operator>] <destPortVal>] [**-nextHop** <nextHopVal>] [**-srcMac** <mac_addr>] [**-protocol** <protocol> | **-protocolNumber** <positive_integer>] [**-vlan** <positive_integer>] [**-interface** <interface_name>] [**-priority** <positive_integer>] [**-msr** (ENABLED | DISABLED) [**-monitor** <string>]] [**-state** (ENABLED | DISABLED)]
- show ns pbr [<name>]

Example

```
> set ns pbr pbr1 -priority 23
 Done
> show ns pbr pbr1
1)      Name: pbr1
        Action: ALLOW                   Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                      Protocol:
        Vlan:                       Interface:
        Active Status: DISABLED          Applied Status: NOTAPPLIED
        Priority: 23
        NextHop: 10.102.29.77
 Done
```

To remove one or all PBRs by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- rm ns pbr <name>
- clear ns PBRs

Example

```
> rm ns pbr pbr1
 Done
> clear ns PBRs
 Done
```

Parameters for configuring a PBR

name

The name (alphanumeric) of the PBR.

action

The action to perform on packets that match the PBR. Possible values: ALLOW, DENY.

nextHop

The IP address of the next hop router or the name of the link load balancing virtual server to which to send matching packets if action is set to ALLOW. If you specify a link load balancing virtual server, which can provide a backup if a next hop link fails, first make sure that the next hops bound to it are actually next hops that are directly connected to the NetScaler appliance. Otherwise, the NetScaler will throw an error when you attempt to create the PBR.

srcIP

IP address of the source machine. You can also specify a range of addresses, by enclosing the low and high addresses in brackets (for example, [10.102.29.50-10.102.29.100]).

operator

You can use the following operators while creating PBRs: = and !=

destIP

The IP address of the destination system. You can also specify a range of addresses, by enclosing the low and high addresses in brackets (for example, [10.102.33.31-10.102.33.100]).

srcPort

The port address of the source system. You also can specify a range of ports, by enclosing the low and high port numbers in brackets (for example [30-90]).

Note: The source port can be modified only for TCP and UDP.

destPort

The port address of the destination system. You also can specify a range of ports, by enclosing the low and high port numbers in brackets (for example [40-90]).

Note: The destination port can be modified only for TCP and UDP.

protocol

The protocol field in the IP header. Possible values: ICMP, IGMP, TCP, EGP, IGP, ARGUS, UDP, RDP, RSVP, EIGRP, L2TP, and ISIS.

protocolNumber

The IP protocol number (decimal). Minimum value: 1. Maximum value: 255.

srcMac

The MAC address of the source system. Only the last 32 bits are considered during a lookup.

vlan

The VLAN ID present in the VLAN tag of the packet. Possible values: 1 to 4094.

interface

This is the network interface on which the packet arrives.

priority

The priority of the ACL. Possible values: 0 to 10240.

state

The state of the PBR. Possible Values: ENABLED, DISABLED. Default: Enabled.

msr

Enable or disable Monitored Static Route(MSR) on this route. This parameter is not applicable if you specify an LLB virtual server name for the nextHop parameter. Possible values: ENABLED, DISABLED Default value: DISABLED.

monitor

The name of the monitor of type PING or ARP.

state

The state of the PBR. Possible values: ENABLED, DISABLED Default value: ENABLED

To create a PBR by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **PBRs**.
2. In the details pane, do one of the following:
   - To create a new PBR, click **Add**.
   - To modify an existing PBR, click **Open**.
3. In the **Create PBR** or **Configure PBR** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a PBR” as shown:

   - **Name***-name
   - **Action**-action
   - **Next Hop**-nextHop
   - **Source, Operator**-operator
   - **Source, Low/High**-srcIP (To specify a specific IP address, type the same address in both text boxes.
   - **Destination, Operator**-operator
   - **Destination, Low/High**-destIP (To specify a specific IP address, type the same address in both text boxes.
   - **Protocol**-protocol (or protocolNumber)
   - **Source Port, Operator**-operator
   - **Source Port, Low/High**-srcPort
   - **Destination Port, Operator**-operator
   - **Destination Port, Low/High**-destPort
   - **Source Mac**-srcMac
   - **VLAN**-vlan
   - **Interface**-interface
   - **Priority**-priority
   - **Enable PBR**-state
   - **Monitored Static Route**-msr
   - **Monitor**-monitor

   * A required parameter

4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the PBR has been configured successfully.

To remove one or all PBRs by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **PBRs**.

2. To remove a single PBR, in the details pane, select the PBR that you want to remove (for example, **p1**), and then click **Remove**.

3. To remove all PBRs, click **Clear**.

Applying a PBR

You must apply a PBR to activate it. The following procedure reapplies all PBRs that you have not disabled. The PBRs constitute a memory tree (lookup table). For example, if you create 10 PBRs (p1 - p10), and then you create another PBR (p11) and apply it, all of the PBRs (p1 - p11) are freshly applied and a new lookup table is created. If a session has a DENY PBR related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR. For example, you must follow this procedure after disabling a PBR.

Note: PBRs created on the NetScaler appliance do not work until they are applied.

To apply a PBR by using the NetScaler command line

At the NetScaler command prompt, type:

apply ns PBRs

To apply a PBR by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **PBRs**.
2. In the details pane, select the PBR that you want to apply (for example, **p1**).
3. Click **Commit**.
4. In the **Apply PBR(s)** dialog box, click **Yes**.

Enabling or Disabling PBRs

By default, the PBRs are enabled. This means that when PBRs are applied, the NetScaler appliance automatically compares incoming packets against the configured PBRs. If a PBR is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBRs are applied. After the PBRs are applied, the NetScaler does not compare incoming packets against disabled PBRs.

To enable or disable a PBR by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable ns pbr <name>
- disable ns pbr <name>

Examples

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                   Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                      Protocol:
        Vlan:                       Interface:
        Active Status: ENABLED              Applied Status: APPLIED
        Priority: 10
        NextHop: 10.102.29.77

 Done

> disable ns PBR pbr1
Warning: PBR modified, use 'apply pbrs' to commit this operation
> apply pbrs
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                   Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                      Protocol:
```

```
    Vlan:                       Interface:
    Active Status: DISABLED         Applied Status: NOTAPPLIED
    Priority: 10
    NextHop: 10.102.29.77
Done
```

To enable or disable a PBR by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **PBRs**.
2. In the details pane, select the PBR (for example, **p1**) and do one of the following:
   - To enable the PBR, click **Enable**.
   - To disable the PBR, click **Disable**.

   A message appears in the status bar, stating that the PBR has been successfully enabled or disabled.

Renumbering PBRs

You can automatically renumber the PBRs to set their priorities to multiples of 10.

To renumber PBRs by using the NetScaler command line

At the NetScaler command prompt, type:

renumber ns pbrs

To renumber PBRs by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **PBRs**.
2. In the details pane, click **Renumber Priority (s)**.
3. In the **Renumber Priority(s) PBR(s)** dialog box, click **Yes**.

Use Case - PBR with Multiple Hops

Consider a scenario in which two PBRs, PBR1 and PBR2, are configured on NetScaler appliance NS1. PBR1 routes all the outgoing packets, with source IP address as 10.102.29.30, to next hop router R1. PBR2 routes all the outgoing packets, with source IP address as 10.102.29.90, to next hop router R2. R3 is another next hop router connected to NS1.

If router R1 fails, all the outgoing packets that matched against PBR1 are dropped. To avoid this situation, you can specify a link load balancing (LLB) virtual server in the next hop field while creating or modifying a PBR. Multiple next hops are bound to the LLB virtual server as services (for example R1, R2, and R3). Now, if R1 fails, all the packets that matched against PBR1 are routed to R2 or R3 as determined by the LB method configured on the LLB virtual server.

The NetScaler appliance throws an error if you attempt to create a PBR with an LLB virtual server as the next hop in the following cases:

- Adding another PBR with the same LLB virtual server.
- Specifying a nonexistent LLB virtual server.
- Specifying an LLB virtual server for which the bound services are not next hops.
- Specifying an LLB virtual server for which the LB method is not set to one of the following:
    - LEASTPACKETS
    - LEASTBANDWIDTH
    - DESTIPHASH
    - SOURCEIPHASH
    - WEIGHTDRR
    - SRCIPDESTIP_HASH
    - LTRM
    - CUSTOM LOAD
- Specifying an LLB virtual server for which the LB persistence type is not set to one of the following:
    - DESTIP
    - SOURCEIP

- SRCDSTIP

The following table lists the names and values of the entities configured on the NetScaler appliance:

Table 1. Sample Values for Creating Entities

Entity Type	Name	IP Address
Link load balancing virtual server	LLB1	NA
Services (next hops)	Router1	1.1.1.1.254
	Router2	2.2.2.2.254
	Router3	3.3.3.3.254
PBRs	PBR1	NA
	PBR2	NA

To implement the configuration described above, you need to:

1. Create services Router1, Router2, and Router3 that represent next hop routers R1, R2, and R3.
2. Create link load balancing virtual server LLB1 and bind services Router1, Router2, and Router3 to it.
3. Create PBRs PBR1 and PBR2, with next hop fields set as LLB1 and 2.2.2.254 (IP address of the router R2), respectively.

To create a service by using the NetScaler command line

At the NetScaler command prompt, type:

- add service <name> <IP> <serviceType> <port>
- show service <name>

Example

```
> add service Router1 1.1.1.1.254 ANY *
 Done
> add service Router2 2.2.2.254 ANY *
 Done
> add service Router3 3.3.3.254 ANY *
 Done
```

Parameters for creating a service

name

The name of the service. Maximum length: 127

IP

The IP address of the physical router for which a service will be added.

serviceType

The type of connections that the service will handle. Specify a service type of ANY.

port

Port on which the service listens. Specify an asterisk (*) as the port number.

To create services by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. 3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service, as shown:
   - **Service Name***—name
   - **Server**—IP
   - **Protocol***—serviceType (Select **ANY** from the drop-down list.)
   - **Port***—port

   * A required parameter
4. Click **Create**.
5. Repeat Steps 2-4 to create another service.
6. Click **Close**.
7. In the **Services** pane, select the services that you just configured and verify that the settings displayed at the bottom of the screen are correct.

To create a link load balancing virtual server and bind a service by using the NetScaler command line

At the NetScaler command prompt, type:

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

Example

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
```

Parameters for creating an LLB virtual server

name

The name of the load balancing virtual server being added. Maximum length: 127

serviceType

The service type. Possible value: ANY.

Parameters for binding the service

name

The virtual server name to which the service is bound. Maximum length: 127

serviceName

The name of the service that is bound. Maximum Length: 127

To create a link load balancing virtual server and bind a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** pane, click **Add**.
3. In the **Create Virtual Servers (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating an LLB virtual server, as shown:
   - **Name***—name
   - **Protocol***—serviceType (Select **ANY**.)

   * A required parameter

   Note: Make sure **Directly Addressable** is unchecked.
4. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.
5. Click **Create**, and then click **Close**.
6. In the **Load Balancing Virtual Servers** tab, select the virtual server that you just created, and verify that the settings displayed in the **Details** pane are correct.

To create a PBR by using the NetScaler command line

At the NetScaler command prompt, type:

- add ns pbr <name> <action> [**-srcIP** [<operator>] <srcIPVal>] [**-nextHop** <nextHopVal>]
- sh ns pbr

Example

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
```

Parameters for configuring a PBR

name

The name (alphanumeric) of the PBR.

action

The action to perform on packets that match the PBR. Possible values: ALLOW, DENY.

nextHop

The IP address of the next hop router or the name of the link load balancing virtual server to which to send matching packets if action is set to ALLOW. If you specify a link load balancing virtual server, which can provide a backup if a next hop link fails, first make sure that the next hops bound to it are actually next hops that are directly connected to the NetScaler appliance. Otherwise, the NetScaler will throw an error when you attempt to create the PBR.

srcIP

IP address of the source machine. You can also specify a range of addresses, by enclosing the low and high addresses in brackets (for example, [10.102.29.50-10.102.29.100]).

To create a PBR by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **PBRs**.
2. In the details pane, do one of the following:
   - To create a new PBR, click **Add**.
   - To modify an existing PBR, click **Open**.
3. In the **Create PBR** or **Configure PBR** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a PBR” as shown:
   - **Name***-name
   - **Action**-action
   - **Next Hop**-nextHop
   - **Source, Operator**-operator
   - **Source, Low/High**-srcIP (To specify a specific IP address, type the same address in both text boxes.

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the PBR has been configured successfully.

Troubleshooting Routing Issues

To make your troubleshooting process as efficient as possible, begin by gathering information about your network. You need to obtain the following information about the NetScaler appliance and other systems in the Network:

- Complete Topology diagram, including interface connectivity and intermediate switch details.
- Running Configuration. You can use the show running command to get the running configuration for ns.conf and ZebOS.conf.
- Output of the History command, to determine whether any configuration changes were made when the issue arose.
- Output of the Top and ps -ax commands, to determine whether any routing daemon is overutilizing the CPU or is misbehaving.
- Any routing related core files in /var/core - nsm, bgpd, ospfd, or ripd. Check the time stamp to see if they are relevant.
- dr_error.log and dr_info.log files from /var/log.
- Output of the date command and time details for all relevant systems. Print dates across all devices one after another, so that the times on the log messages can be correlated with various events.
- Relevant ns.log, newnslog files.
- Configuration files, log files and command history details from upstream and downstream routers.

Generic Routing FAQs

Users typically have the following questions about how to troubleshoot generic routing issues:

- How do I enable Health Monitoring for content switching virtual servers?

  By default, the states of content switching virtual servers are not updated. Therefore, these servers always remain up, which makes RHI ineffective. Use the nsapimgr knob to enable updating CS vserver states.

  root@ns# nsapimgr -y -s csw_state_update=1

- How do I save the config files?

  The write command from VTYSH saves only ZebOS.conf. Run the save config command from NetScaler CLI to save both ns.conf and ZebOS.conf files.

- If I have configured both a static default route and a dynamically learned default route, which is the preferred default route?

  The dynamically learned route is the preferred default route. This behavior is unique to default routes. However, in case of the Network Services Module (NSM), unless the administrative distances are modified, a statically configured route in the RIB is preferred over a dynamic route. The route that is downloaded to the NSM FIB is the static route.

- How do I block the advertisement of default routes?

  After release 7.0, the default route is not injected into ZebOS.

  However, if you are working with 7.0 or an earlier release, you must apply a suitable route map with the

  However, if you are working with 7.0 or an earlier release, you must apply a suitable route map with the redistribute kernel command for each protocol to block default route advertisement. For example:

  ns(config)#access-list 1 deny 0.0.0.0
  ns(config)#access-list 2 permit any
  ns(config)#route-map redist-kernel permit 5

```
ns(config-route-map)#match ip address 1
ns(config)#route-map redist-kernel permit 10
ns(config-route-map)#match ip address 2
ns(config-route-map)#q
ns(config)#router ospf 1
ns(config-router)#redistribute kernel route-map redist-kernel
ns(config-router)#q
ns(config)#q
ns#show route-map
route-map redist-kernel, permit, sequence 5
  Match clauses:
    ip address 1
  Set clauses:
route-map redist-kernel, permit, sequence 10
  Match clauses:
    ip address 2
  Set clauses:
ns#show access-list
Standard IP access list 1
    deny 0.0.0.0
Standard IP access list 2
    permit any
ns#
```

- How do I view the debug output of networking daemons?

  You can write debugging output from networking daemons to a file by entering the following log file command from the global configuration view in VTYSH:

  ```
  ns(config)#log file /var/ZebOS.log
  ```

  With release 8.1, you can direct debug output to the console by entering the terminal monitor command from VTYSH user view:

  ```
  ns#terminal monitor
  ```

- How do I collect cores of running daemons?

  You can use the gcore utility to collect cores of running daemons for processing by gdb. This might be helpful in debugging misbehaving daemons without bringing the whole routing operation to a standstill.

  ```
  gcore [-s] [-c core] [executable] pid
  ```

  The **-s** option temporarily stops the daemon while gathering the core image. This is a recommended option, because it guarantees that the resulting image shows the core in a consistent state.

root@ns#gcore -s -c nsm.core /netscaler/nsm 342

- How do I run a batch of ZebOS commands?

  You can run a batch of ZebOS commands from a file by entering the VTYSH -f <file-name> command. This does not replace the running configuration, but appends to it. However, by including commands to delete the existing configuration in the batch file and then add those for the new, desired configuration, you can use this mechanism to replace a specific configuration:

```
!
router bgp 234
network 1.1.1.1 255.255.255.0
!
route-map bgp-out2 permit 10
 set metric 9900
 set community 8602:300
!
```

Troubleshooting OSPF-Specific Issues

Before you start debugging any OSPF specific issue, you must collect information from the NetScaler appliance and all systems in the affected LAN, including upstream and downstream routers. To begin, enter the following commands:

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route
6. show ip ospf database summary
    a. If there are only few LSAs in the database, then enter show ip ospf database router, show ip ospf database A. network, show ip ospf database external, and other commands to get the full details of LSAs.
    b. If there are a large number of LSAs in the database, enter the show ip ospf database self-originated command.
7. show ip ospf
8. show ns ip. This ensures that the details of all VIPs of interest are included.
9. Get the logs from peering devices and run the following command:

    gcore -s -c xyz.core /netscaler/ospfd <pid>

Note: The gcore command is non-disruptive.

Collect additional information from the NetScaler as follows:

1. Enable logging of error messages by entering the following command from the global configuration view in VTYSH:

    ns(config)#log file /var/ospf.log

2. Get the details of:

    ./nsconmsg -g ospf

3. For adjacency related defects, run the following command:

./nsapimgr -B "call nsospf_print_area"

Note: This command is not supported in NetScaler 9.3 nCore.

4. Enable debugging ospf events and log them by using the following command:

   ns(config)#log file /var/ospf.log

   Enable debug ospf lsa packet only if the number of LSAs in the database is relatively small (< 500).

Internet Protocol version 6 (IPv6)

A NetScaler appliance supports both server-side and client-side IPv6 and can therefore function as an IPv6 node. It can accept connections from IPv6 nodes (both hosts and routers) and from IPv4 nodes, and can perform Protocol Translation (RFC 2765) before sending traffic to the services. You have to license the IPv6 feature before you can implement it.

The following table lists some of the IPv6 features that the NetScaler appliance supports.

Table 1. Some Supported IPv6 Features

IPv6 features
IPv6 addresses for SNIPs (NSIP6, VIP6, and SNIP6)
Neighbor Discovery (Address Resolution, Duplicated Address Detection, Neighbor Unreachability Detection, Router Discovery)
Management Applications (ping6, telnet6, ssh6)
Static Routing and Dynamic routing (OSPF)
Port Based VLANs
Access Control Lists for IPv6 addresses (ACL6)
IPv6 Protocols (TCP6, UDP6, ICMP6)
Server Side Support (IPv6 addresses for vservers, services)
USIP (Use source IP) and DSR (Direct Server Return) for IPv6
SNMP and CVPN for IPv6
HA with native IPv6 node address
IPv6 addresses for MIPs
Path-MTU discovery for IPv6

The following table lists NetScaler components that support IPv6 addresses and provides references to the topics that document the components.

Table 2. NetScaler Components That Support IPv6 Addresses and the Corresponding Documentation

NetScaler component	Topic that documents IPv6 support
Network	Adding, Customizing, Removing, Removing all, and Viewing routes.
SSL Offload	Creating IPv6 vservers for SSL Offload
SSL Offload	Specifying IPv6 SSL Offload Monitors
SSL Offload	Creating IPv6 SSL Offload Servers

Load Balancing	Creating IPv6 vservers for Load Balancing
Load Balancing	Specifying IPv6 Load Balancing Monitors
Load Balancing	Creating IPv6 Load Balancing Servers
DNS	Creating AAAA Records

You can configure IPv6 support for the above features after implementing the IPv6 feature on your NetScaler appliance. You can configure both tagged and prefix-based VLANs for IPv6. You can also map IPv4 addresses to IPv6 addresses.

Implementing IPv6 Support

IPv6 support is a licensed feature, which you have to enable before you can use or configure it. If IPv6 is disabled, the NetScaler does not process IPv6 packets. It displays the following warning when you run an unsupported command:

"Warning: Feature(s) not enabled [IPv6PT]"

The following message appears if you attempt to run IPv6 commands without the appropriate license:

"ERROR: Feature(s) not licensed"

After licensing the feature, use either of the following procedures to enable or disable IPv6.

To enable or disable IPv6 by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable ns feature ipv6pt
- disable ns feature ipv6pt

To enable or disable IPv6 by using the configuration utility

1. In the navigation pane, expand **System** and click **Settings**.
2. On the **Settings** page, under **Modes and Features**, click **change advanced features**.
3. In the **Configure Advanced Features** dialog box, do one of the following:
   - To enable IPv6, select the **IPv6 Protocol Translation** check box.
   - To disable IPv6, clear the **IPv6 Protocol Translation** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** dialog box, click **Yes**.

VLAN Support

If you need to send broadcast or multicast packets without identifying the VLAN (for example, during DAD for NSIP, or ND6 for the next hop of the route), you can configure the NetScaler appliance to send the packet on all the interfaces with appropriate tagging. The VLAN is identified by ND6, and a data packet is sent only on the VLAN.

For more information about ND6 and VLANs, see Configuring Neighbor Discovery.

Port-based VLANs are common for IPv4 and IPv6. Prefix-based VLANs are supported for IPv6.

Simple Deployment Scenario

Following is an example of a simple load balancing set-up consisting of an IPv6 vserver and IPv4 services, as illustrated in the following topology diagram.

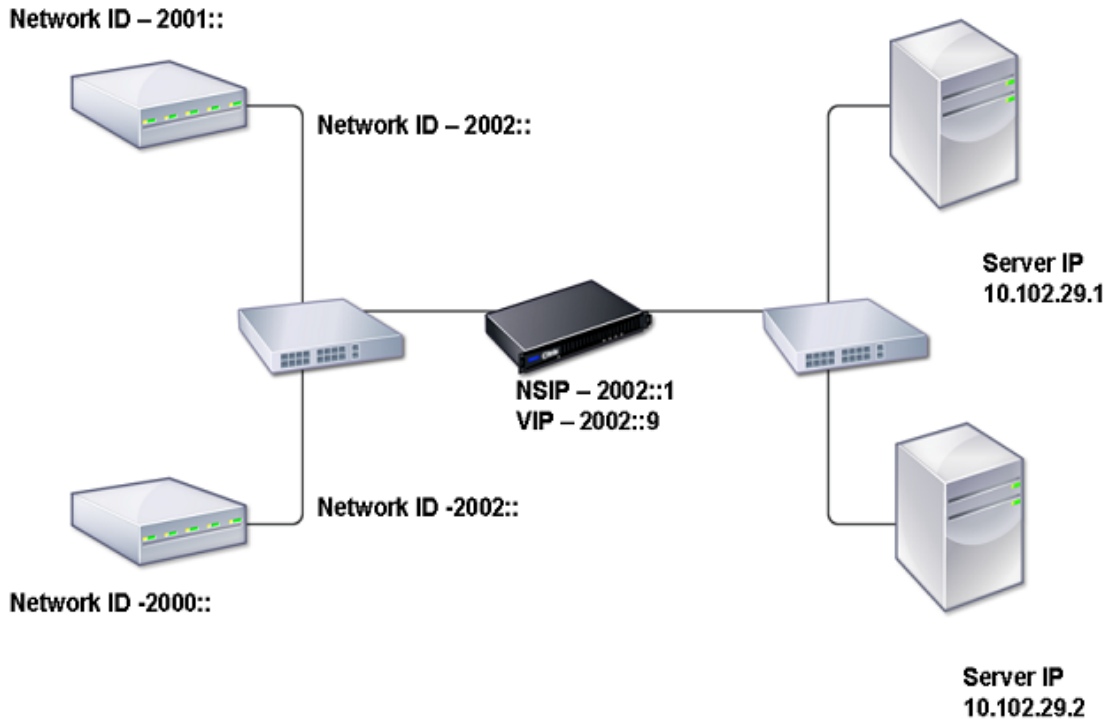
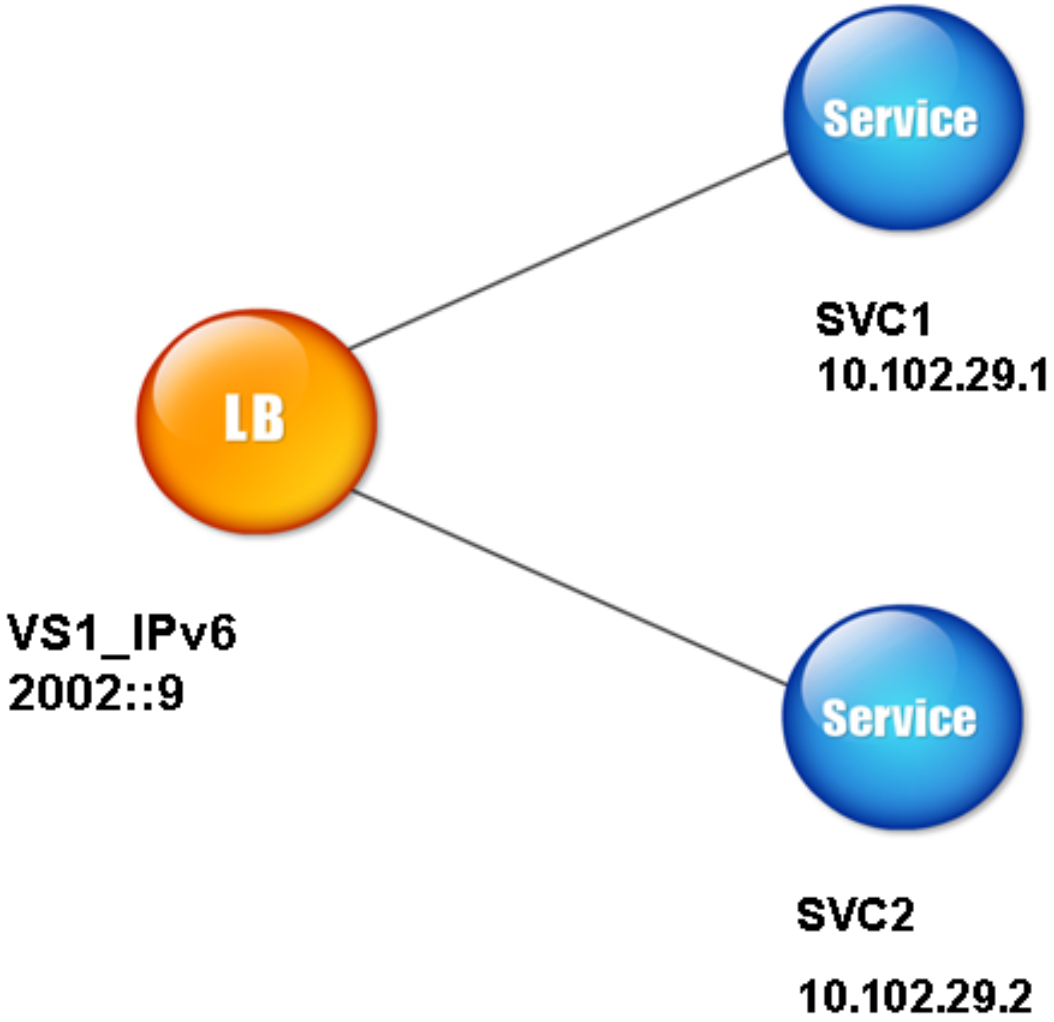


Figure 1. IPv6 Sample Topology

The following table summarizes the names and values of the entities that must be configured on the NetScaler.

Table 1. Sample Values for Creating Entities

Entity type	Name	Value
LB Vserver	VS1_IPv6	2002::9
Services	SVC1	10.102.29.1
	SVC2	10.102.29.2



The following figure shows the entities and values of the parameters to be configured on the NetScaler.Figure 2. IPv6 Entity Diagram

To configure this deployment scenario, you need to do the following:

1. Create an IPv6 service.
2. Create an IPv6 LB vserver.
3. Bind the services to the vserver.

To create IPv4 services by using the NetScaler command line

At the NetScaler command prompt, type:

add service <Name> <IPAddress> <Protocol> <Port>

Example

add service SVC1 10.102.29.1 HTTP 80
add service SVC2 10.102.29.2 HTTP 80

To create IPv4 services by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Services**.
2. On the **Services** page, click **Add.**
3. In the **Create Service** dialog box, in the **Service Name**, **Server**, and **Port** text boxes, type the name, IP address, and port of the service (for example, SVC1, 10.102.29.1, and 80).
4. In the **Protocol** drop-down list box, select the type of the service (for example, **HTTP**).
5. Click **Create** and click **Close.**
6. Repeat Steps 1-5 to create a service SVC2 with IP address 10.102.29.2 and port 80.

To create IPv6 vserver by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <Name> <IPAddress> <Protocol> <Port>

Example

add lb vserver VS1_IPv6 2002::9 HTTP 80

To create IPv6 vserver by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** page, click **Add.**
3. In the **Create Virtual Servers (Load Balancing)** dialog box, select the **IPv6** check box.
4. In the **Name**, **Port**, and **IP Addresses** text boxes, type the name, port, and IP address of the vserver (for example, VS1_IPv6, 80, and 2002::9).
5. Click **Create** and click **Close.**

To bind a service to an LB vserver by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <name> <service>

Example

bind lb vserver VS1_IPv6 SVC1

The vservers receive IPv6 packets and the NetScaler performs Protocol Translation (RFC 2765) before sending traffic to the IPv4-based services.

To bind a service to an LB vserver by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** page, select the vserver for which you want to bind the service (for example, VS1_IPv6).
3. Click **Open.**
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box corresponding to the service that you want to bind to the vserver (for example, SVC1).
5. Click **OK**.
6. Repeat Steps 1-4 to bind the service (for example, SVC2 to the vserver).

Host Header Modification

When an HTTP request has an IPv6 address in the host header, and the server does not understand the IPv6 address, you must map the IPv6 address to an IPv4 address. The IPv4 address is then used in the host header of the HTTP request sent to the vserver.

To change the IPv6 address in the host header to an IPv4 address by using the NetScaler command line

At the NetScaler command prompt, type:

set ns ip6 <IPv6Address> **-map** <IPAddress>

Example

set ns ip6 2002::9 -map 200.200.200.200

To change the IPv6 address in the host header to an IPv4 address by using the configuration utility

1. In the navigation pane, expand **Networks** and click **IPs**.
2. In the **IPs** page, click the **IPV6s** tab and select the IP address for which you want to configure a mapped IP address, for example, 2002:0:0:0:0:0:0:9.
3. Click **Open.**
4. In the **Configure IP6** dialog box, in the **Mapped IP** text box, type the mapped IP address that you want to configure, for example, 200.200.200.200.
5. Click **OK.**

VIP Insertion

If an IPv6 address is sent to an IPv4-based server, the server may not understand the IP address in the HTTP header, and may generate an error. To avoid this, you can map an IPv4 address to the IPv6 VIP and enable VIP insertion.

To configure a mapped IPv6 address by using the NetScaler command line

At the NetScaler command prompt, type:

set ns ip6 <IPv6Address> **-map** <IPAddress>

Example

```
> set ns ip6 2002::9 -map 200.200.200.200
 Done
```

To configure a mapped IPv6 address by using the configuration utility

1. In the navigation pane, expand **Networks** and click **IPs**.
2. In the **IPs** page, click the **IPV6s** tab and select the IP address for which you want to configure a mapped IP address (for example, 2002:0:0:0:0:0:0:9).
3. Click **Open**.
4. In the **Configure IP6** dialog box, in the **Mapped IP** text box, type the mapped IP address that you want to configure (for example, 200.200.200.200).
5. Click **OK.**

Use either of the following procedures to enable insertion of an Ipv4 VIP address and port number in the HTTP requests sent to the servers.

To enable VIP insertion by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <name> -insertVserverIPPort <Value>

Example

```
> set lb vserver VS1_IPv6 -insertVserverIPPort ON
 Done
```

To enable VIP insertion by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** page, in the **Load Balancing Virtual Servers** page, select the vserver that you want to enable port insertion (for example, VS1_IPv6).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
5. In the **Vserver IP Port Insertion** drop-down list box, select **VIPADDR.**
6. In the **Vserver IP Port Insertion text box**, type the vip header.

Cloud Bridge

A fundamental part of the Citrix® Cloud framework, the Citrix NetScaler® Cloud Bridge™ feature is a tool used to build a cloud-extended data center. The Cloud Bridge enables you to connect one or more cloud computing instances—virtual servers in the cloud—to your network without reconfiguring your network. Cloud-hosted applications appear as though they are running on one contiguous enterprise network.

The primary purpose of the Cloud Bridge is to enable companies to move their applications to the cloud while reducing costs and the risk of application failure. In addition, the Cloud Bridge increases network security in cloud environments.

A Cloud Bridge is an extension of bridge (layer 2) that connects a NetScaler appliance or virtual appliance residing in a cloud to a NetScaler appliance or virtual appliance on your LAN. The connection is made through an IP tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets, from a wide variety of network protocols, to be forwarded over another protocol. GRE is used to:

- Connect networks running non-IP, non-routable protocols, such as Appletalk, Novell IPX, and NetBIOS.
- Bridge across a wide area network (WAN).
- Create a transport tunnel for any type of traffic that needs to be sent unchanged across a different network.

The GRE protocol encapsulates packets by adding a GRE and a GRE IP header to the packets.

Cloud Bridge supports use of the open-standard Internet Protocol security (IPSec) protocol suite to secure communications between peers in the Cloud Bridge.

In an Cloud Bridge, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

The IPSec use the transport mode in which only the payload of the GRE-encapsulated packet is encrypted. The encryption uses the Encapsulating Security Payload (ESP) protocol, which ensures the integrity of the packet by using a HMAC hash function and ensures confidentiality by using an encryption algorithm. The ESP protocol after encrypting the payload and calculating the HMAC, generates an ESP header and then inserts it after the GRE IP header. The ESP protocol also generates an ESP trailer and then inserts it at the end of the encrypted packets.

Before securing the communication between the peers in the Cloud Bridge, using the Internet Key Exchange (IKE) protocol in IPSec:

1. The two peers mutually authenticate with each other, using one of the following authentication methods:

    - **Pre-shared key authentication**. A text string called a pre-shared key, manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.

    - **Digital certificates authentication.** For digital certificate authentication, a peer (sender) initiator signs message interchange data by using its private key, and the other peer(receiver) uses the peer(sender's) public key to verify the signature. Typically, the public key is exchanged in messages containing an X.509v3 certificate. This certificate provides a level of assurance that a peer's identity as represented in the certificate is associated with a particular public key.
2. The peers then negotiate to reach agreement on:

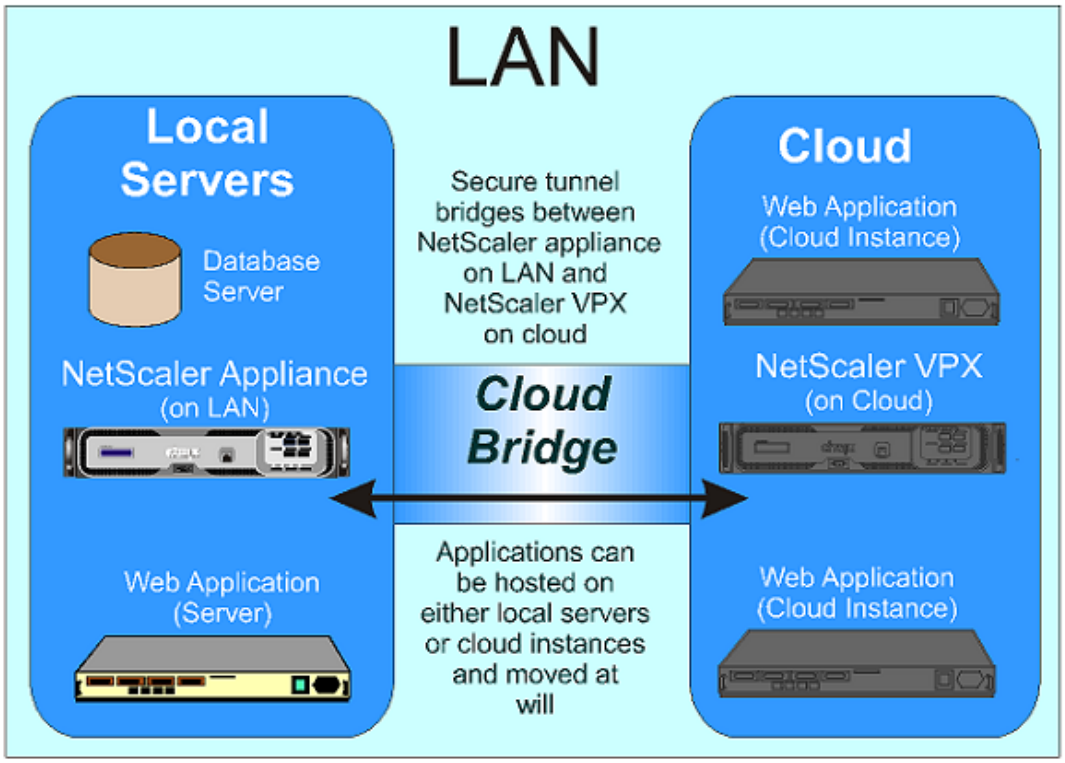
    - A security protocol to use, so that each one sends data in a format the other can understand

    - An encryption algorithm

    - Cryptographic keys for encrypting data in one peer and decrypting the data in the other

This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one way (simplex). For example, when two NetScaler appliances, NS1 and NS2, are communicating by means of IPSec over a Cloud Bridge, NS1 has two Security Associations. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets.

SAs expire after a specified interval of time, which is called the lifetime. The two peers then using the Internet Key Exchange (IKE) protocol (part of the IPSec protocol suite) negotiates new cryptographic keys and establishes new SAs. The purpose of the limited lifetime is to prevent attackers from cracking a key.



Following is a conceptual illustration of an Cloud Bridge.Figure 1. Conceptual Diagram-NetScaler Cloud Bridge

About the Cloud Bridge

A OpenCloud Bridge is a Layer-2 network bridge that connects a virtual appliance on a cloud instance to a NetScaler appliance on your LAN. The connection is made through a tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. GRE is used to:

- Connect networks running non-IP, nonroutable protocols, such as Appletalk, Novell IPX, and NetBIOS.
- Bridge across a wide area network (WAN).
- Create a transport tunnel for any type of traffic that needs to be sent unchanged across a different network.

The GRE protocol encapsulates packets by adding a GRE and a GRE IP header to the packets.

OpenCloud Bridge supports the use of open standard Internet Protocol security (IPSec) protocol suite to secure the communication between the peers in the Cloud bridge.

In an OpenCloud bridge, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

The IPSec use the transport mode in which only the payload of the GRE encapsulates packet is encrypted. The encryption is done by the Encapsulating Security Payload (ESP) protocol. The ESP protocol ensures the integrity of the packet using a HMAC hash function and the confidentiality using a encryption algorithm. After encrypting the payload and calculating the HMAC, an ESP header is generated and is inserted after the GRE IP header and a ESP trailer is inserted at the end of the encrypted payload. An Authentication Header (AH) is also added before the ESP header for data origin authentication of the packet.

Cloud bridge also supports the NAT implementation of RFC 3947 and 3948 for the cloud bridge peers to communicate properly when any of the peer is behind a NAT device.

Before securing the communication between the peers in the OpenCloud bridge, using the IKE protocol in IPSec:

- The two peers mutually authenticate with each other. One of the following authentication methods is used:

- **Pre-shared key authentication**. A text string called pre-shared key is manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication. Therefore for the authentication to be successful, you must configure the same pre-shared key on each of peer.

- **Digitital certificates authentication**. For digital certificate authentication, a peer (sender) initiator signs message interchange data using his private key, and the other peer(receiver) uses the peer(sender's) public key to verify the signature. Typically, the public key is exchanged via messages containing an X.509v3 certificate. This certificate provides a level of assurance that a peer's identity-as represented in the certificate-is associated with a particular public key.

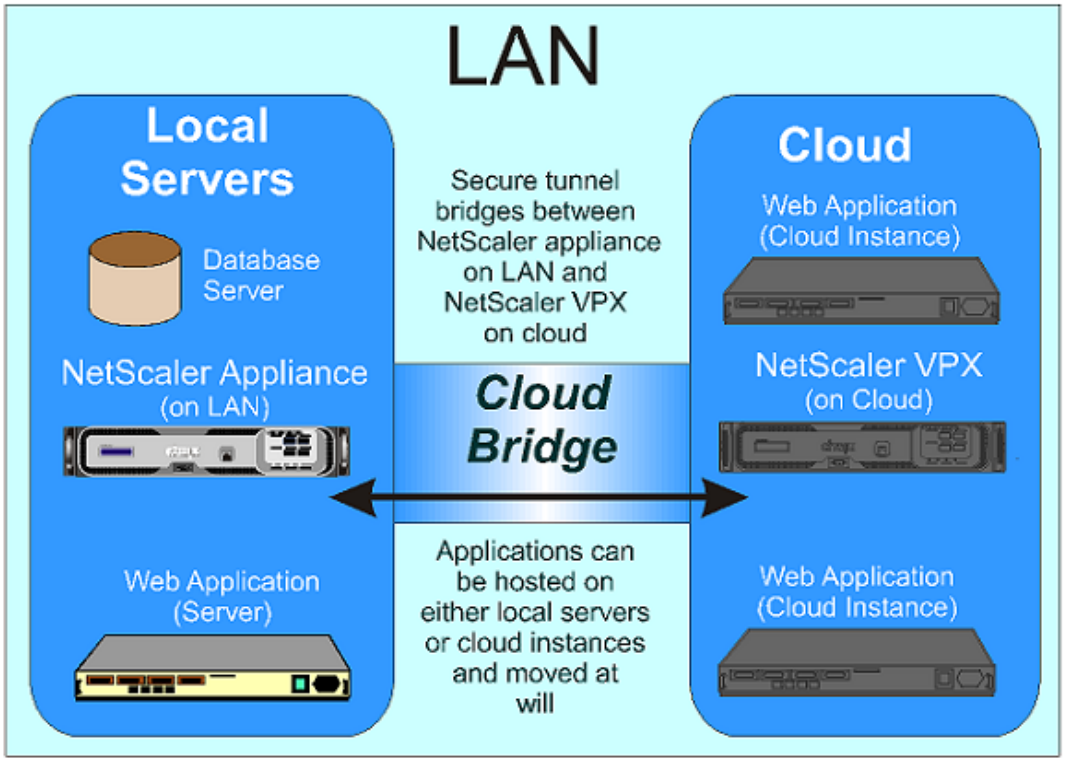
- The peers then agree on the:

  - security protocol to use, so that each one sends data in a format the other can understand.

  - encryption algorithm and negotiate cryptographic keys for encrypting data in one peer and decrypting the data in the other.

This agreement of the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one way (simplex). For example, when two NetScalers, NS1 and NS2, are communicating using IPSec in an OpenCloud Bridge, then the NS1 will have two Security Associations. One SA is used for processing out-bound packets and other SA is used for processing inbound packets.

SAs expire after a specified interval of time called the lifetime. Using the IKE protocol in IPSec, new SAs are established where new cryptographic keys were negotiated between the peers of the OCB. This prevents the cracking of the key by an attacker.



Following is a conceptual illustration of an OpenCloud Bridge.Figure 1. Conceptual Diagram of LAN That Uses the NetScaler OpenCloud Bridge

Setting Up a Cloud Bridge - Method 1

Before setting up a Cloud Bridge, you must configure the NetScaler appliance or VPX virtual appliance on the LAN and the appliance or virtual appliance on the Cloud.

To configure a new NetScaler appliance, see Getting Started with Citrix NetScaler. To configure a new virtual appliance, see Getting Started with Citrix NetScaler VPX.

You must then configure networking on both appliances. Each of the two configurations may include a VLAN that contains the servers or the cloud instances that will use the Cloud Bridge. To configure VLANs, see Configure a VLAN.

To set up a Cloud Bridge, on the NetScaler appliance or virtual appliance that anchors the LAN side of the Cloud Bridge:

1. Configure a GRE tunnel.
2. Configure IPSec on the GRE tunnel.
3. Configure a Cloud Bridge:
    - Create a logical representation of the Cloud Bridge by specifying a name.
    - Bind one or more GRE tunnels to the Cloud Bridge.
    - Bind VLANs and IP addresses to the Cloud Bridge (Optional.)

You then repeat these steps on the NetScaler appliance or virtual appliance that anchors the cloud side of the Cloud Bridge.

You can perform these tasks individually (Method 1), or you can configure everything in one dialog box in dialog box in the configuration utility (Method 2). For more information, see Setting Up a Cloud Bridge – Method 2.

Creating IP Tunnels

For enabling IP/IP (IP Tunneling) for a specific virtual IP (VIP) address you need to create an IP tunnel manually, known as configured tunnels. For Cloud Bridge, you need to create GRE tunnels.

To create an IP tunnel by using the NetScaler command line

At the NetScaler command prompt type:

- add iptunnel <name> <remoteIp> <remoteSubnetMask> <localIp> **-type** -**protocol** (ipoverip | GRE) -**secure** (YES | NO)

- show iptunnel

To modify or remove an IP tunnel by using the NetScaler command line

- To modify an IP tunnel, type the set iptunnel command, the name of the tunnel, and the parameters to be changed, with their new values.
- To remove an IP tunnel, type the rm iptunnel command and the name of the tunnel.

Parameters for creating an IP tunnel

name

Name of the IP Tunnel. This alphanumeric string is required and cannot be changed after the service group is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

remoteIp

A public IPv4 address of the remote NetScaler appliance used to set up the tunnel.

remoteSubnetMask

Subnet mask of the remote IP address of the tunnel.

localIp

A public IPv4 address of the local NetScaler appliance used to set up the tunnel. Possible values: Auto, MIP, SNIP, and VIP. Default: Auto.

protocol

The protocol to be used in setting up the IP tunnel. Select GRE for using the Generic Routing Encapsulation (GRE) protocol to set up a GRE tunnel.

secure

Enable or disable IPSec for securing communication in the GRE tunnel.

To create an IP Tunnel by using the configuration utility

1. In the navigation pane, expand **Network**, and click **IP Tunnels**.
2. In the details pane, click **Add**.
3. In the **Add IP Tunnel** dialog box, specify values for the following parameters:
   - **Name*—**name
   - **Remote IP*—**remoteIp
   - **Remote Mask*—**remoteSubnetMask
   - **Local IP Type***—localIp (in the **local IP Type** drop down list, select one of the IP type (Mapped IP, Subnet IP, and Virtual). All the configured IPs of the selected IP type will be populated in the **Local IP** drop down list. Select the desired IP from the list.)
   - **Protocol—**protocol
   - **Secure—**secure

   *A required parameter.
4. Click **Create**, and then click **Close.**

Configuring IPSec on a GRE tunnel

For configuring IPSec on a GRE tunnel:

- The **Secure** parameter should be enabled on the GRE tunnel.
- You need to specify the same local IP address and the remote IP address that you specified for the GRE tunnel.

To configure IPSec on a GRE tunnel by using the NetScaler command line

At the NetScaler command prompt, type:

add ipsec peer <name> <localIP> <peerIP> [**-encAlgo** (AES | 3DES) ...] [**-hashAlgo** <hashAlgo> ...] [**-lifetime** <positive_integer>] (**-psk** |(**-publickey** <string> **-privatekey** <string> **-peerPublicKey** <string>))

To modify or remove an IPSec config by using the NetScaler command line

- To modify an IPSec config, type the set ipsec peer command, the name of the IPSec config, and the parameters to be changed, with their new values.
- To remove an IPSec config, type the rm ipsec peer command and the name of the IPSec config.

Parameters for configuring IPSec on a GRE tunnel

name

A name for an IPSec configuration. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

localIP

Local IPv4 address to be used for security association. This parameter should be same as the local IPv4 or IPv6 address that was used to set up a GRE tunnel.

peerIP

The IPv4 address of the peer for security association. This parameter should be same as the IPv4 address of the remote NetScaler appliance or VPX virtual appliance that was used to set up the GRE tunnel.

encAlgo

The encryption algorithm to be used in IPSec configuration for a Cloud Bridge. Possible values: AES, 3DES.

hashAlgo

The encryption algorithm to be used in IPSec configuration for a Cloud Bridge. Possible values: HMAC_SHA1, HMAC_SHA256, HMAC_MD5. Default: HMAC_SHA1.

lifetime

Time, in seconds, after which the security association expires. After expiration, new SAs are established, and new cryptographic keys are negotiated between the peers connected by the Cloud Bridge. Maximum value: 31536000. Default: 28800.

psk

A text string, called the pre-shared key, to be manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication before security associations are established. Therefore, for the authentication to be successful, you must configure the same pre-shared key on both of the peers of the Cloud Bridge. Maximum Length: 63 characters.

publickey

A local digital certificate to be used to authenticate the local NetScaler appliance to the remote peer before establishing IPSec security associations. The same certificate should be present and set for the Peer Public Key parameter in the remote peer.

privatekey

The private key of the local digital certificate.

peerPublicKey

A digital certificate of the remote peer. This certificate is used to authenticate the remote peer to the local peer before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the remote peer.

To configure IPSec on a GRE tunnel by using the configuration utility

1. In the navigation pane, expand **Cloud Bridge**, and then click **IPSec Peer**.
2. In the details pane, click **Add**.
3. In the **Create IPSec Peer** dialog box, type or select values for the following parameters, which correspond to parameters described in "Parameters for configuring IPSec" as shown:
   - **Name***—name
   - **IP Address***—peerIP
   - **IP Address***—localIP
   - **Encryption Algorithm**—encAlgo
   - **Hash Algorithm**—hashAlgo
   - **Lifetime**—lifetime
   - **Pre-Shared key Exists**—psk
   - **Public Key**—publickey
   - **Private Key**—privatekey
   - **Peer Public Key**—peerPublicKey

   * A required parameter.
4. Click **Create**, and then click **Close**.

Configuring a Cloud Bridge

You can think of the Cloud Bridge as a group that holds a set of secure GRE tunnels. After configuring GRE tunnels secured with IPSec, you need to create a logical representation of the Cloud Bridge by assigning a name to a Cloud Bridge and binding one or more configured GRE tunnels to the Cloud Bridge. You can then bind VLANs and IP subnets to the new Cloud Bridge. The VLAN and IP subnet settings are common to all the GRE tunnels bound to the Cloud Bridge.

To create a cloud bridge by using the NetScaler command line

At the NetScaler command prompt, type:

add netbridge <name>

To bind GRE tunnels, VLANs, and IP Subnets to a Cloud Bridge by using the NetScaler command line

At the NetScaler command prompt, type:

bind netbridge <name> [**-tunnel** <name>] [**-vlan** <id>] [**-IPAddress** <ip_addr|ipv6_addr>]

To modify or remove an Cloud Bridge by using the NetScaler command line

- To modify a Cloud Bridge, type the set netbridge command, the name of the Cloud Bridge, and the parameters to be changed, with their new values.
- To remove a Cloud Bridge, type the rm netbridge command and the name of the Cloud Bridge.

Parameters for configuring a Cloud Bridge

name

The name of the Cloud Bridge that you are configuring. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to tell which NetScaler appliances the Cloud Bridge connects.

tunnel

The name of the GRE tunnel to be associated with the Cloud Bridge.

VLAN

The ID of the local VLAN that needs to be extended to the cloud.

IPAddress

The IPV4 subnet that needs to be extended to the cloud.

To configure a Cloud Bridge by using the configuration utility

1. In the navigation pane, expand **Cloud Bridge**, and then click **Network Bridge**.
2. In the details pane, do one of the following:
   - To create a new Cloud Bridge, click **Add**.
   - To modify an existing Cloud Bridge, select the Cloud Bridge, and then click **Open**.
3. In the **Create Network Bridge** dialog box, type a name for your new Cloud Bridge.
4. In the **Create Network Bridge** or **Configure Network Bridge** dialog box, on the **Tunnels** tab (selected by default), do one of the following to bind GRE tunnels to the Cloud Bridge:
   - If the GRE tunnels that you want are listed, select the corresponding check boxes.
   - If you want bind all the GRE tunnels listed, click **Activate All**.
   - If you want to create a new GRE tunnel, click **Add**.
5. In the **Create Network Bridge** or **Configure Network Bridge** dialog box, on the **VLANs** tab (selected by default), do one of the following to bind GRE tunnels to the Cloud Bridge:
   - If the VLANS that you want are listed, select the corresponding check boxes.
   - If you want bind all the VLANs listed, click **Activate All**.
   - If you want to create a new VLAN, click **Add**.
6. On the **IP Subnets** tab, do the following to bind IP subnets to the Cloud Bridge:
   - If you want to bind a new IP subnet, click **Add**.
   - If you want to modify an existing IP subnet, click **Open**.
7. Click **Create**, and then click **Close**.

Setting Up Cloud Bridge-Method 2

For configuring a Cloud Bridge, you need to perform the following steps on each of the appliances that is to a be peer of the Cloud Bridge.

1. Configure a GRE tunnel.
2. Configure IPSec on the GRE tunnel.
3. Create a logical representation of the Cloud Bridge by specifying a name.
4. Bind one or more GRE Tunnel to the Cloud Bridge.
5. Bind VLANs and IP addresses to the Cloud Bridge (Optional.)

The configuration utility provides a single dialog box on which you can perform all of these steps to configure a Cloud Bridge.

When you use this dialog box:

- A GRE tunnel, IPSec, and Cloud Bridge entities are created with the same name.
- The GRE tunnel created is configured with IPSec.

By using this method, you can configure a Cloud Bridge with only one GRE tunnel. You can later modify the Cloud Bridge to bind more GRE tunnels to it. For more information, see Configuring a Network Bridge.

Parameters for configuring a Cloud Bridge

name

The name of the Cloud Bridge that you are configuring. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to know which NetScaler appliances the Cloud Bridge connects.

Local IP

A public IPv4 address of the local NetScaler appliance or VPX virtual appliance. This address is used to set up a GRE tunnel, with IPSec configuration, to a public IP IPv4 address of the remote peer NetScaler appliance or virtual appliance.

Remote IP

A public IPv4 address of the remote peer NetScaler appliance or virtual appliance. This is the address that is used to at the remote peer to set up the GRE tunnel, with IPSec configuration, with the local peer.

Pre-Shared key

A text string, called the pre-shared key, to be manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication before security associations are established. Therefore, for the authentication to be successful, you must configure the same pre-shared key on both of the peers of the Cloud Bridge. Maximum Length: 63 characters.

Public key

A local digital certificate to be used to authenticate the local NetScaler appliance to the remote peer before establishing IPSec security associations. The same certificate should be present and set for the Peer Public Key parameter in the remote peer.

Private Key

The private key of the local digital certificate.

Peer Public Key

A digital certificate of the remote peer. This certificate is used to authenticate the remote peer to the local peer before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the remote peer.

To configure a Cloud Bridge by using the configuration utility

1. In the navigation pane, click **Cloud Bridge**.
2. In the details pane, under Getting Started, click **Configure Cloud Bridge**.
3. In the **Configure Cloud Bridge** dialog box, specify values for the following parameters, which are described in "Parameters for configuring a Cloud Bridge":
   - **Name***
   - **Local IP***
   - **Remote IP***

   * A required parameter.
4. Do one of the following to select an IPSec authentication method between the peers for establishing IPSec security associations:
   - For pre-shared key authentication method, select **Pre-shared Key** and specify values for the following parameters, which are described in "Parameters for configuring a Cloud Bridge":
     - **Pre-Shared key**
     - **Confirm Key**

     * A required parameter.
   - For digital certificates authentication method, select **Certificate** and specify values for the following parameters, which are described in "Parameters for configuring a Cloud Bridge":
     - **Public Key**
     - **Private Key**
     - **Peer Public Key**

     * A required parameter.
5. Click **Create**, and then click **Close**.

Setting Up Cloud Bridge to SoftLayer Enterprise Cloud

The configuration utility includes a wizard that helps you to easily configure a Cloud Bridge between a NetScaler appliance on any network and NetScaler VPX instances on the SoftLayer enterprise cloud.

Note: This feature is supported only on NetScaler 9.3.e.

Using the wizard you can perform the following steps to configure a Cloud Bridge to a NetScaler VPX instance on the SoftLayer enterprise cloud.

1. Connect to the SoftLayer enterprise cloud by providing the user-login credentials.
2. Select the Citrix XenServer that is running the NetScaler VPX appliance.
3. Select the NetScaler VPX appliance.
4. Provide Cloud Bridge parameters for:
    - Configuring a GRE Tunnel.
    - Configuring IPSec on the GRE tunnel.
    - Creating a logical representation of the Cloud Bridge by specifying a name.
    - Binding the GRE Tunnel to the Cloud Bridge.

When you use this wizard, a GRE tunnel, IPSec, and Cloud Bridge entities are created each on both the peers.

To configure a Cloud Bridge by using the configuration utility

1. In the navigation pane, click **Cloud Bridge**.
2. In the details pane, click **SOFTLAYER**.
3. In the **Setup CloudBridge on SoftLayer wizard**, click **Next**, and then follow the instructions in the wizard.

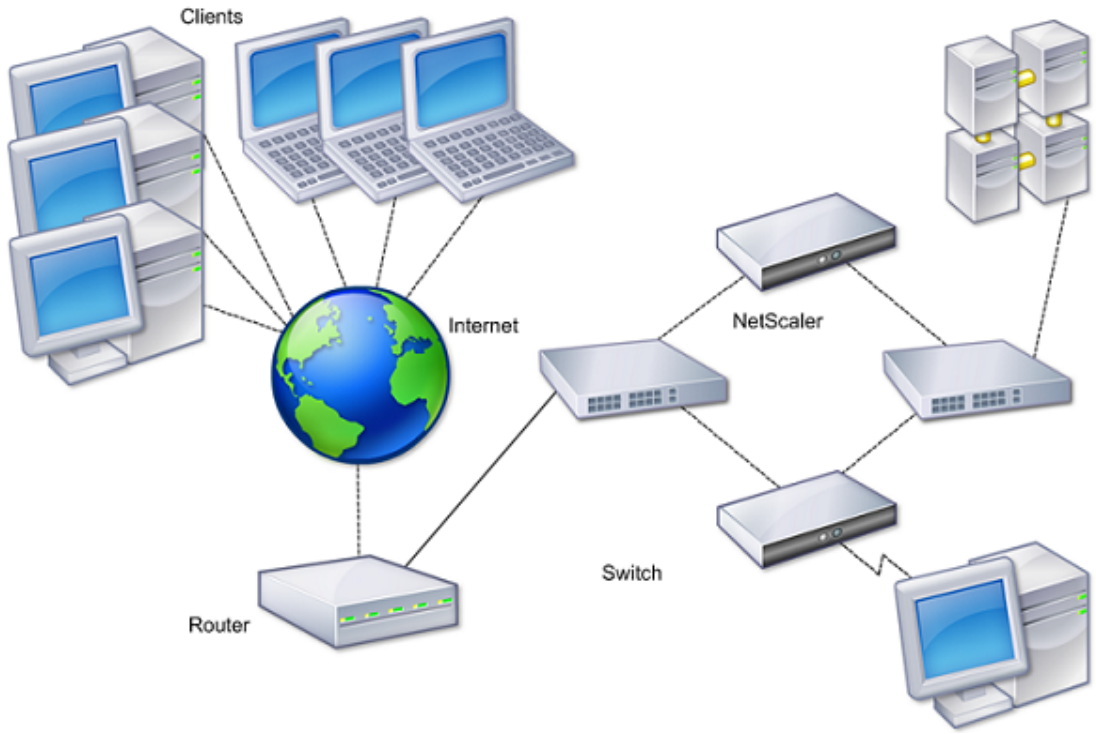
High Availability

A high availability (HA) deployment of two Citrix® NetScaler® appliances can provide uninterrupted operation in any transaction. With one appliance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If, for any reason, the primary node is unable to accept connections, the secondary node takes over.

The secondary node monitors the primary by sending periodic messages (often called heartbeat messages or health checks) to determine whether the primary node is accepting connections. If a health check fails, the secondary node retries the connection for a specified period, after which it determines that the primary node is not functioning normally. The secondary node then takes over for the primary (a process called failover).

After a failover, all clients must reestablish their connections to the managed servers, but the session persistence rules are maintained as they were before the failover.

With Web server logging persistence enabled, no log data is lost due to the failover. For logging persistence to be enabled, the log server configuration must carry entries for both systems in the log.conf file.



The following figure shows a network configuration with an HA pair.Figure 1. NetScaler Appliances in a High Availability Configuration

To configure HA, you might want to begin by creating a basic setup, with both nodes in the same subnet. You can then customize the intervals at which the nodes communicate health-check information, the process by which nodes maintain synchronization, and the

propagation of commands from the primary to the secondary. You can configure fail-safe mode to prevent a situation in which neither node is primary. If your environment includes devices that do not accept NetScaler gratuitous ARP messages, you should configure virtual MAC addresses. When you are ready for a more complex configuration, you can configure HA nodes in different subnets.

To improve the reliability of your HA setup, you can configure route monitors and create redundant links. In some situations, such as when troubleshooting or performing maintenance tasks, you might want to force a node to fail over (assign primary status to the other node), or you might want to force the secondary node to stay secondary or the primary node to stay primary.

Considerations for a High Availability Setup

Note the following requirements for configuring systems in an HA setup:

- In an HA configuration, the primary and secondary NetScaler appliances should be of the same model. Different NetScaler models are not supported in an HA pair (for example, you cannot configure a 10010 model and a 7000 model as an HA pair).
- In an HA setup, both nodes must run the same version of NetScaler, for example, nCore/nCore or classic/classic. If the nodes are running NetScaler classic and you want to migrate to NetScaler nCore of the same NetScaler release, prop and sync are not supported during the migration process. Once migration is complete, prop and sync are auto-enabled. The same applies if you migrate from NetScaler nCore to NetScaler classic.
- Entries in the configuration file (ns.conf) on both the primary and the secondary system must match, with the following exceptions:
  - The primary and the secondary systems must each be configured with their own unique NetScaler IP addresses (NSIPs.)
  - In an HA pair, the node ID and associated IP address of one node must point to the other node. For example, if you have nodes NS1 and NS2, you must configure NS1 with a unique node ID and the IP address of NS2, and you must configure NS2 with a unique node ID and the IP address of NS1.
- If you create a configuration file on either node by using a method that does not go directly through the GUI or the CLI (for example, importing SSL certificates, or changing to startup scripts), you must copy the configuration file to the other node or create an identical file on that node.
- Initially, all NetScaler appliances are configured with the same RPC node password. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. For security, you should change the default RPC node passwords.

  One RPC node exists on each NetScaler. This node stores the password, which is checked against the password provided by the contacting system. To communicate with other systems, each NetScaler requires knowledge of those systems, including how to authenticate on those systems. RPC nodes maintain this information, which includes the IP addresses of the other systems, and the passwords they require for authentication.

  RPC nodes are implicitly created when adding a node or adding a Global Server Load Balancing (GSLB) site. You cannot create or delete RPC nodes manually.

  Note: If the NetScaler appliances in a high availability setup are configured in one-arm mode, you must disable all system interfaces except the one connected to the switch or hub.

- For an IPv6 HA configuration, the following considerations apply:
    - You must install the IPv6PT license on both NetScaler appliances.
    - After installing the IPv6PT license, enable the IPv6 feature by using the configuration utility or the NetScaler command line.
    - Both NetScaler appliances require a global NSIP IPv6 address. In addition, network entities (for example, switches and routers) between the two nodes must support IPv6.

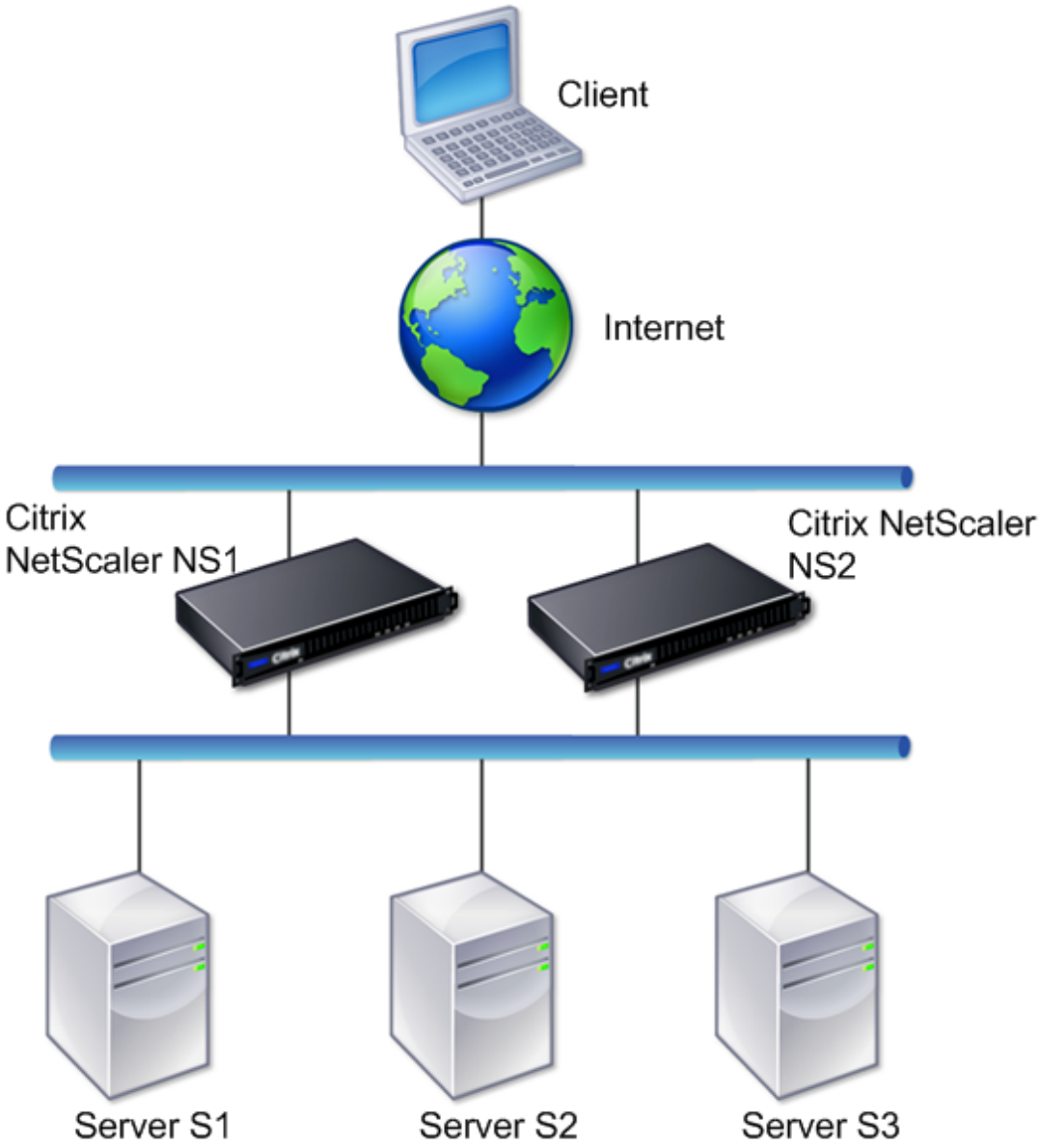
Configuring High Availability

To set up a high availability configuration, you create two nodes, each of which defines the other's NetScaler IP (NSIP) address as a remote node. Begin by logging on to one of the two NetScaler appliances that you want to configure for high availability, and add a node. Specify the other appliance's NetScaler IP (NSIP) address as the address of the new node. Then, log on to the other appliance and add a node that has the NSIP address of the first appliance. An algorithm determines which node becomes primary and which becomes secondary.

Note: The configuration utility provides an option that avoids having to log on to the second appliance.

The following figure shows a simple HA setup, in which both nodes are in same subnet.

Figure 1. Two NetScaler Appliances Connected in a High Availability Configuration


Client
Internet
Citrix
NetScaler NS1
Citrix NetScaler
NS2
Server S1
Server S2
Server S3

Adding a Remote Node

To add a remote NetScaler appliance as a node in a high availability setup, you specify a unique node ID and the appliance's NSIP. The maximum number of node IDs in an HA setup is 64. When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

Note: To ensure that each node in the high availability configuration has the same settings, you should synchronize your SSL certificates, startup scripts, and other configuration files with those on the primary node.

To add a node by using the NetScaler command line

At the NetScaler command prompt, type:

- add ha node <id> <IPAddress>
- sh ha node

Example

```
> add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
> sh ha node
```

To disable an HA monitor by using the NetScaler command line

At the NetScaler command prompt, type:

- set interface <ifNum> [**-haMonitor** (**ON** | **OFF**)]
- show interface <ifNum>

Example

```
> set interface 1/3 -haMonitor OFF
 Done
> show interface 1/3
```

```
        Interface 1/3 (Fast Ethernet 10/100 MBits) #5
        flags=0x4000 [ENABLED, DOWN, down, autoneg, 802.1q]
        MTU=1514, native vlan=5, MAC=00:d0:68:0b:58:dc, downtime 332h55m50s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl ON,
              throughput 0

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
```

Parameters for adding a remote node

node id

Unique number that identifies the node to be added. Possible values: 1 to 64.

IPAddress

IPv4 or IPv6 address of the node to be added.

id

Interface number, in slot/port notation.

haMonitor

Monitor the specified interface for failing events. Possible values: ON, OFF. Default: ON.

To add a remote node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, select the **Nodes** tab, and then click **Add**.
3. In the **High Availability Setup** dialog box, in the **Remote Node IP Address** text box, type the NSIP address of the NetScaler that is to be added as the remote node. If the NSIP is an IPv6 address, select the **IPv6** check box before entering the address.
4. If you want to add the local node to the remote node automatically, select the **Configure remote system to participate in High Availability setup** check box. If you do not select this option, you will have to log in to the appliance represented by the remote node and add the node that you are currently configuring.
5. Make sure that the **Turn off HA monitor on interfaces/channels that are down** check box is selected.
6. Click **OK**. The **Nodes** page displays both of the nodes in your HA configuration (the local node and the remote node).

Disabling or Enabling a Node

You can disable or enable only a secondary node. When you disable a secondary node, it stops sending heartbeat messages to the primary node, and therefore the primary node can no longer check the status of the secondary. When you enable a node, the node takes part in the high availability configuration.

To disable or enable a node by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- set ha node -hastatus DISABLED
- set ha node -hastatus ENABLED

To disable or enable a node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open.**
3. In the **Configure Node** dialog box, under **High Availability Status**, do one of the following:
   - To enable the node, select the **DISABLED (Do not participate in HA)** check box.
   - To enable the node, select the **ENABLED (Do not participate in HA)** check box.
4. Click **OK**. A message appears in the status bar, stating that the node has been configured successfully.

Removing a Node

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the NetScaler command line

At the NetScaler command prompt, type:

rm ha node <id>

Example

```
> rm ha node 2
  Done
```

To remove a node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. On the **High Availability** page, select the **Nodes** tab.
3. On the **Nodes** page, select the node that you want to remove, and click **Remove**.
4. On the **Remove** dialog box, click **Yes.**

Note: You can use the Network Visualizer to view the NetScaler appliances that are configured as a high availability (HA) pair and perform high availability configuration tasks. For more information, see Using the Network Visualizer.

Configuring the Communication Intervals

The hello interval is the interval at which the heartbeat messages are sent to the peer node. The dead interval is the time interval after which the peer node is marked DOWN if heartbeat packets are not received. The heartbeat messages are UDP packets sent to port 3003 of the other node in an HA pair.

To set the hello and dead intervals by using the NetScaler command line

At the NetScaler command prompt, type:

- set HA node [-**helloInterval** <msecs>] [-**deadInterval** <secs>]
- show HA node [<id>]

Parameters for setting the hello and dead intervals

helloInterval

Interval between successive heartbeat messages, in milliseconds. Possible values: 200 to 1000. Default: 200.

deadInterval

Number of seconds after which a node is marked DOWN if there is no response to heartbeat messages. Possible values: 3 to 60. Default: 3.

To set the hello and dead intervals by using the configuration utility

1. In the navigation pane, expand **System** and click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open.**
3. In the **Configure Node** dialog box, under **Intervals**, specify values for the following parameters, which correspond to parameters described in “Parameters for setting the hello and dead intervals” as shown:
   - **Hello Interval (msecs)**—helloInterval
   - **Dead Interval (secs)**—deadInterval
4. Click **OK**. A message appears in the status bar, stating that the node has been configured successfully.

Configuring Synchronization

Synchronization is a process of duplicating the configuration of the primary node on the secondary node. The purpose of synchronization is to ensure that there is no loss of configuration information between the primary and the secondary nodes, regardless of the number of failovers that occur. Synchronization uses port 3010.

Synchronization is triggered by either of the following circumstances:

- The secondary node in an HA setup comes up after a restart.
- The primary node becomes secondary after a failover.

Automatic synchronization is enabled by default. You can also force synchronization.

Disabling or Enabling Synchronization

Automatic HA synchronization is enabled by default on each node in an HA pair. You can enable or disable it on either node.

To disable or enable automatic synchronization by using the NetScaler command line

At the NetScaler command prompt, type:

- set HA node -haSync DISABLED
- set HA node -haSync ENABLED

To disable or enable synchronization by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open**.
3. In the **Configure Node** dialog box, under **HA Synchronization**, do one of the following:
   - To disable HA synchronization, clear the **Secondary node will fetch the configuration from Primary** check box.
   - To enable HA synchronization, select the **Secondary node will fetch the configuration from Primary** check box.
4. Click **OK**. A message appears in the status bar, stating that the node has been configured successfully.

Forcing the Secondary Node to Synchronize with the Primary Node

In addition to automatic synchronization, the NetScaler supports forced synchronization. You can force the synchronization from either the primary or the secondary node. When you force synchronization from the secondary node, it starts synchronizing its configuration with the primary node.

However, if synchronization is already in progress, forced synchronization fails and the system displays a warning. Forced synchronization also fails in any of the following circumstances:

- You force synchronization on a standalone system.
- The secondary node is disabled.
- HA synchronization is disabled on the secondary node.

To force synchronization by using the NetScaler command line

At the NetScaler command prompt, type:

force HA sync

To force synchronization by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, click **Force Synchronization**.

Synchronizing Configuration Files in a High Availability Setup

In a high availability setup, you can synchronize various configuration files from the primary node to the secondary node.

To perform the synchronization, you can use the NetScaler command line or the configuration utility at either the primary or the secondary node. Files located on the secondary that are specific to the secondary (not present on the primary) are not deleted during the synchronization.

To synchronize files in a high availability setup by using the NetScaler command line

At the NetScaler command prompt, type:

sync HA files <mode>

Example

```
> sync HA files all
 Done
```

Parameters for synchronizing files in a high availability setup

Mode

The type of synchronization to be performed. Following are descriptions of the available options. Each description includes, in parentheses, the command-line argument that specifies that option.

- **Everything except licenses and rc.conf** (all). Synchronizes files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, and Application Firewall XML objects. Synchronization paths:
  - /nsconfig/ssl/

    - /var/netscaler/ssl/
    - /var/vpn/bookmarks/
    - /nsconfig/htmlinjection/
    - /nsconfig/monitors/
    - /nsconfig/nstemplates/
    - /nsconfig/rc.netscaler
    - /nsconfig/inetd.conf
    - /nsconfig/sshd_config
    - /nsconfig/hosts
    - /nsconfig/snmpd.conf
    - /nsconfig/ntp.conf
    - /nsconfig/resolv.conf
    - /nsconfig/syslog.conf
- **Bookmarks** (bookmarks). Synchronizes all Access Gateway bookmarks. Synchronization path:
    - /var/vpn/bookmark/
- **SSL certificates and keys** (ssl). Synchronizes all certificates, keys, and CRLs for the SSL feature. Synchronization paths:
    - /nsconfig/ssl/
    - /var/netscaler/ssl/
- **EdgeSight Monitoring (HTML injection) scripts** (htmlinjection). Synchronizes all scripts configured for the HTML injection feature. Synchronization path:
    - /nsconfig/htmlinjection/
- **Imported XML objects** (imports). Synchronizes all XML objects (for example, WSDLs, schemas, error pages) configured for the Application Firewall. Synchronization path:
    - /var/download/
- **Licenses and rc.conf** (misc). Synchronizes all license files and the rc.conf file. Synchronization paths:
    - /nsconfig/license/
    - /nsconfig/rc.conf
- **Everything including licenses and rc.conf** (all_plus_misc). Synchronizes files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, Application Firewall XML objects, licenses, and the rc.conf file. Synchronization paths:
    - /nsconfig/ssl/

- /var/netscaler/ssl/
- /var/vpn/bookmarks/
- /nsconfig/htmlinjection/
- /nsconfig/monitors/
- /nsconfig/nstemplates/
- /nsconfig/rc.netscaler
- /nsconfig/inetd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/resolv.conf
- /nsconfig/syslog.conf
- /nsconfig/license/
- /nsconfig/rc.conf

To synchronize files in a high availability setup by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Diagnostics**.
2. In the details pane, under **Utilities**, click **Start file synchronization**.
3. In the **Start file synchronization** dialog box, in the **Mode** drop-down list, select the appropriate type of synchronization (for example, **Everything except licenses and rc.conf**), and then click **OK**.

Configuring Command Propagation

In an HA setup, any command issued on the primary node propagates automatically to, and is executed on, the secondary before it is executed on the primary. If command propagation fails, or if command execution fails on the secondary, the primary node executes the command and logs an error. Command propagation uses port 3011.

In an HA pair configuration, command propagation is enabled by default on both the primary and secondary nodes. You can enable or disable command propagation on either node in an HA pair. If you disable command propagation on the primary node, commands are not propagated to the secondary node. If you disable command propagation on the secondary node, commands propagated from the primary are not executed on the secondary node.

Note: After reenabling propagation, remember to force synchronization.

If synchronization occurs while you are disabling propagation, any configuration-related changes that you make before the disabling of propagation takes effect are synchronized with the secondary node. This is also true for cases where propagation is disabled while synchronization is in progress.

To disable or enable command propagation by using the NetScaler command line

At the NetScaler command prompt, type:

- set HA node -haProp ENABLED
- set HA node -haProp ENABLED

To disable or enable command propagation by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open**.
3. In the **Configure Node** dialog box, under **HA Propagation**, do one of the following:
   - To disable HA Propagation, clear the **Primary node will propagate configuration to the Secondary** check box.
   - To enable HA Propagation, select the **Primary node will propagate configuration to the Secondary** check box.
4. Click **OK**. A message appears in the status bar, stating that the node has been configured successfully.

Configuring Fail-Safe Mode

In an HA configuration, fail-safe mode ensures that one node is always primary when both nodes fail the health check. This is to ensure that when a node is only partially available, backup methods are enabled to handle traffic as best as possible. The HA fail-safe mode is configured independently on each node.

The following table shows some of the fail-safe cases. The NOT_UP state means that the node failed the health check yet it is partially available. The UP state means that the node passed the health check.

Table 1. Fail-Safe Mode Cases

Node A (Primary) Health State	Node B (Secondary) Health State	Default HA Behavior	Fail-Safe Enabled HA Behavior	Description
NOT_UP(failed last)	NOT_UP (failed first)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
NOT_UP (failed first)	NOT_UP(failed last)	A (Secondary), B (Secondary)	A(Secondary), B(Primary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
UP	UP	A (Primary), B (Secondary)	A (Primary), B (Secondary)	If both nodes pass the health check, no change in behavior with fail-safe enabled.
UP	NOT_UP	A(Primary), B(Secondary)	A (Primary), B (Secondary)	If only the secondary node fails, no change in behavior with fail-safe enabled.
NOT_UP	UP	A(Secondary), B(Primary)	A(Secondary), B(Primary)	If only the primary fails, no change in behavior with fail-safe enabled.
NOT_UP	UP (STAYSECONDARY)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If the secondary is configured as STAYSECONDARY, the primary remains primary even if it fails.

To enable fail-safe mode by by using the NetScaler command line

At the NetScaler command prompt, type:

set HA node [**-failSafe** (**ON** | **OFF**)]

Example

set ha node -failsafe ON

To enable fail-safe mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open.**
3. In the **Configure Node** dialog box, under **Fail-Safe Mode**, select the **Maintain one Primary node even when both nodes are unhealthy** check box.
4. Click **OK**. A message appears in the status bar, stating that the node has been configured successfully.

Configuring Virtual MAC Addresses

A Virtual MAC address (VMAC) is a floating entity shared by the primary and the secondary nodes in an HA setup.

In an HA setup, the primary node owns all of the floating IP addresses, such as the MIPs, SNIPs, and VIPs. The primary node responds to Address Resolution Protocol (ARP) requests for these IP addresses with its own MAC address. As a result, the ARP table of an external device (for example, an upstream router) is updated with the floating IP address and the primary node's MAC address.

When a failover occurs, the secondary node takes over as the new primary node. It then uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it acquired from the primary. However, the MAC address that the new primary advertises is the MAC address of its own interface.

Some devices (notably a few routers) do not accept the GARP messages generated by the NetScaler appliance. As a result, some external devices retain the old IP to MAC mapping advertised by the old primary node. This can result in a site going down.

You can overcome this problem by configuring a VMAC on both nodes of an HA pair. Both nodes then possess identical MAC addresses. Therefore, when failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

To create a VMAC, you need to first create a Virtual Router ID (VRID) and bind it to an interface. (In an HA setup, you need to bind the VRID to the interfaces on both nodes.) Once the VRID is bound to an interface, the system generates a VMAC with the VRID as the last octet.

Configuring IPv4 VMACs

When you create a IPv4 VMAC address and bind it a interface, any IPv4 packet sent from the interface uses the VMAC address that is bound to the interface. If there is no IPv4 VMAC bound to an interface, the interface's physical MAC address is used.

The generic VMAC is of the form 00:00:5e:00:01:<VRID>. For example, if you create a VRID with a value of 60 and bind it to an interface, the resulting VMAC is 00:00:5e:00:01:3c, where 3c is the hex representation of the VRID. You can create 255 VRIDs with values from 1 to 255.

Creating or Modifying an IPv4 VMAC

You create an IPv4 virtual MAC by assigning it a virtual router ID. You can then you bind the VMAC to an interface. You cannot bind multiple VRIDs to the same interface. To verify the VMAC configuration, you should display and examine the VMACs and the interfaces bound to the VMACs.

To add a VMAC by using the NetScaler command line

At the NetScaler command prompt, type:

- add vrID <id>
- bind vrid <id> **-ifnum** <interface_name>
- sh vrID

Example

```
add vrID 100
bind vrid 100 -ifnum 1/1 1/2 1/3
sh vrID 100
```

To unbind interfaces from a VMAC by using the NetScaler command line

At the NetScaler command prompt, type:

- unbind vrid <id> **-ifnum** <interface_name>
- sh vrID

Parameters for configuring a VMAC

VrID

The VRID that identifies the VMAC. Possible values: 1 to 255.

ifnum

The interface number (slot/port notation) to be bound to the VMAC.

To configure a VMAC by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **VMAC**.
2. In the details pane, on the **VMAC** tab, do one of the following:
   - To create a new VMAC, click **Add**.
   - To modify an existing VMAC, click **Open**.
3. In the **Create VMAC** or **Configure VMAC** dialog box, specify values for the following parameter, which correspond to parameter described in “Parameters for configuring a VMAC” as shown:
   - **Virtual Router ID***—VrID
4. Under **Associate Interfaces**, do one of the following:
   - To bind interfaces to the VMAC, select the desired interfaces from the **Available Interfaces** table, and click **Add**.
   - To unbind interfaces from the VMAC, select the desired interfaces from the **Configured Interfaces** table, and click **Remove**.
5. Click **OK**. A message appears in the status bar, stating that the VMAC has been configured successfully.

Removing an IPv4 VMAC

To remove an IPv4 virtual MAC, you delete its virtual router ID.

To remove an IPv4 VMAC by using the NetScaler command line

At the NetScaler command prompt, type:

rm vrid <id>

Example

rm vrid 100s

To remove an IPv4 VMAC by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **VMAC.**
2. In the details pane, on the **VMAC** tab, select the virtual router ID that you want to remove, and then click **Remove**. A message appears in the status bar, stating that the VMAC has been successfully removed.

Configuring IPv6 VMAC6s

The NetScaler supports VMAC6 for IPv6 packets. You can bind any interface to a VMAC6, even if an IPv4 VMAC is bound to the interface. Any IPv6 packet sent from the interface uses the VMAC6 bound to that interface. If there is no VMAC6 bound to an interface, an IPv6 packet uses the physical MAC.

Creating or Modifying a VMAC6

You create an IPv6 virtual MAC by assigning it an IPv6 virtual router ID. You can then you bind the VMAC to an interface. You cannot bind multiple IPv6 VRIDs to an interface. To verify the VMAC6 configuration, you should display and examine the VMAC6s and the interfaces bound to the VMAC6s.

To add a VMAC6 by using the NetScaler command line

At the NetScaler command prompt, type:

- add vrID6 <id>
- bind vrID6 <id> **-ifnum** <interface_name>
- sh vrID6

Example

```
add vrID6 100
bind vrID6 100 -ifnum 1/1 1/2 1/3
sh vrID6 100
```

To unbind interfaces from a VMAC6 by using the NetScaler command line

At the NetScaler command prompt, type:

- unbind vrID6 <id> **-ifnum** <interface_name>
- sh vrID6

Parameters for configuring a VMAC6

vrID6

The VRID that identifies the VMAC6. Possible values: 1 to 255.

ifnum

The interface number (slot/port notation) to be bound to the VMAC6.

To configure a VMAC6 by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **VMAC**.
2. In the details pane, on the **VMAC6** tab, do one of the following:
   - To create a new VMAC6, click **Add**.
   - To modify an existing VMAC6, click **Open**.
3. In the **Create VMAC6** or **Configure VMAC6** dialog box, specify values for the following parameter, which correspond to parameter described in “Parameters for configuring a VMAC6” as shown:
   - **Virtual Router ID6***—vrID6
4. Under **Associate Interfaces**, do one of the following:
   - To bind interfaces to the VMAC6, select the desired interfaces from the **Available Interfaces** table, and click **Add**.
   - To unbind interfaces from the VMAC6, select the desired interfaces from the **Configured Interfaces** table, and click **Remove**.
5. Click **OK**. A message appears in the status bar, stating that the VMAC6 has been configured successfully.

Removing a VMAC6

To remove an IPv4 virtual MAC, you delete its virtual router ID.

To remove a VMAC6 by using the NetScaler command line

At the NetScaler command prompt, type:

rm vrid6 <id>

Example

rm vrid6 100s

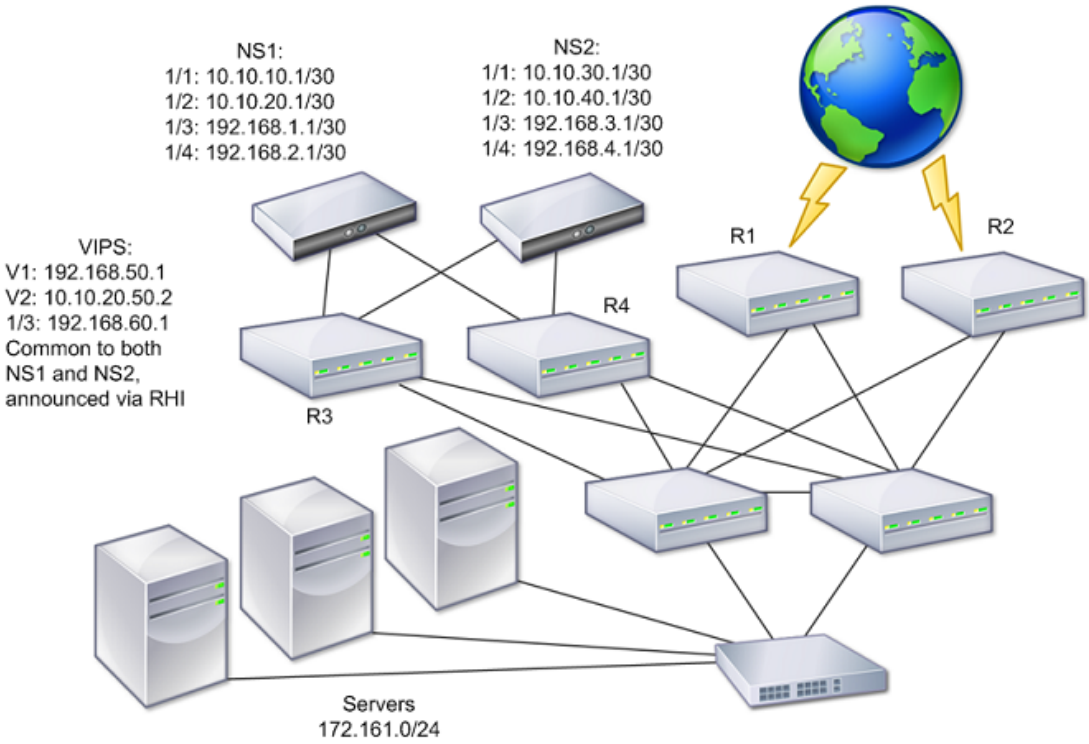
To remove a VMAC6 by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **VMAC.**
2. In the details pane, on the **VMAC6** tab, select the virtual router ID that you want to remove, and then click **Remove**. A message appears in the status bar, stating that the VMAC6 has been successfully removed.

Configuring High Availability Nodes in Different Subnets

The following figure shows an HA deployment with the two systems located in different subnets:

Figure 1. High Availability over a Routed Network



In the figure, the systems NS1 and NS2 are connected to two separate routers, R3 and R4, on two different subnets. The NetScaler appliances exchange heartbeat packets through the routers. This configuration could be expanded to accommodate deployments involving any number of interfaces.

Note: If you use static routing on your network, you must add static routes between all the systems to ensure that heartbeat packets are sent and received successfully. (If you use dynamic routing on your systems, static routes are unnecessary.)

If the nodes in an HA pair reside on two separate networks, the primary and secondary node must have independent network configurations. This means that nodes on different networks cannot share entities such as MIPs, SNIPs, VLANs, and routes. This type of configuration, where the nodes in an HA pair have different configurable parameters, is known as Independent Network Configuration (INC) or Symmetric Network Configuration (SNC).

The following table summarizes the configurable entities and options for an INC, and shows how they must be set on each node.

Table 1. Behavior of NetScaler Entities and Options in an Independent Network Configuration

NetScaler entities	Options
IPs (NSIP/MIP/SNIPs)	Node-specific. Active only on that node.
VIPs	Floating.
VLANs	Node-specific. Active only on that node.
Routes	Node-specific. Active only on that node. Link load balancing routes are floating.
ACLs	Floating (Common). Active on both nodes.
Dynamic routing	Node-specific. Active only on that node. The secondary node should also run the routing protocols and peer with upstream routers.
L2 mode	Floating (Common). Active on both nodes.
L3 mode	Floating (Common). Active on both nodes.
Reverse NAT (RNAT)	Node-specific. RNAT with VIP, because NATIP is floating.

As in configuring HA nodes in the same subnet, to configure HA nodes in different subnets, you log on to each of the two NetScaler appliances and add a remote node representing the other appliance.

Adding a Remote Node

When two nodes of an HA pair reside on different subnets, each node must have a different network configuration. Therefore, to configure two independent systems to function as an HA pair, you must specify INC mode during the configuration process.

When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

To add a node by using the NetScaler command line

At the NetScaler command prompt, type:

- add ha node <id> <IPAddress> **-inc** ENABLED
- sh ha node

Example

```
add ha node 3 10.102.29.170 -inc ENABLED
add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
sh ha node
```

To disable an HA monitor by using the NetScaler command line

At the NetScaler command prompt, type:

- set interface <ifNum> [**-haMonitor** (**ON** | **OFF**)]
- show interface <ifNum>

Example

```
> set interface 1/3 -haMonitor OFF
 Done
> show interface 1/3
        Interface 1/3 (Fast Ethernet 10/100 MBits) #5
        flags=0x4000 (ENABLED, DOWN, down, autoneg, 802.1q)
```

```
        MTU=1514, native vlan=5, MAC=00:d0:68:0b:58:dc, downtime 332h55m50s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl ON,
              throughput 0

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
```

Parameters for adding a remote node

node id

Unique number that identifies the node to be added. Possible values: 1 to 64.

IPAddress

IPv4 or IPv6 address of the node to be added.

inc

Option to be enabled when the nodes that you want to configure for high availability are in different subnets.

id

Interface number, in slot/port notation.

haMonitor

Monitor the specified interface for failing events. Possible values: ON, OFF. Default: ON.

To add a remote node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, select the **Nodes** tab, and then click **Add**.
3. In the **High Availability Setup** dialog box, in the **Remote Node IP Address** text box, type the NSIP address of the NetScaler that is to be added as the remote node. If the NSIP is an IPv6 address, select the **IPv6** check box before entering the address.
4. If you want to add the local node to the remote node automatically, select the **Configure remote system to participate in High Availability setup** check box. If you do not select this option, you will have to log in to the appliance represented by the remote node and add the node that you are currently configuring.
5. Make sure that the **Turn off HA monitor on interfaces/channels that are down** check box is selected.
6. Select the **Turn on INC (Independent Network Configuration) mode on self mode** check box.
7. Click **OK**. The **Nodes** page displays both of the nodes in your HA configuration (the local node and the remote node).

Removing a Node

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the NetScaler command line

At the NetScaler command prompt, type:

rm ha node <id>

Example

```
> rm ha node 2
  Done
```

To remove a node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. On the **High Availability** page, select the **Nodes** tab.
3. On the **Nodes** page, select the node that you want to remove, and click **Remove**.
4. On the **Remove** dialog box, click **Yes.**

Note: You can use the Network Visualizer to view the NetScaler appliances that are configured as a high availability (HA) pair and perform high availability configuration tasks. For more information, see Using the Network Visualizer.

Configuring Route Monitors

You can use route monitors to make the HA state dependent on the internal routing table, whether or not the table contains any dynamically learned or static routes. In an HA configuration, a route monitor on each node watches the internal routing table to make sure that a route entry for reaching a particular network is always present. If the route entry is not present, the state of the route monitor changes to DOWN.

When a NetScaler appliance has only static routes for reaching a network, and you want to create a route monitor for the network, you must enable monitored static routes (MSR) for the static routes. MSR removes unreachable static routes from the internal routing table. If MSR is disabled on static routes, an unreachable static route can remain in the internal routing table, defeating the purpose of having the route monitor.

Route Monitors are supported both in non-INC and INC mode.

Note: Route Monitors in non-INC HA is supported only on NetScaler 9.3.e.

Route Monitors in HA in non-INC mode	Route Monitors in HA in INC mode
Route monitors are propagated by nodes and exchanged during synchronization.	Route monitors are neither propagated by nodes nor exchanged during synchronization.
Route monitors are active only in the current primary node.	Route monitors are active on both the primary and the secondary node.
The NetScaler appliance always displays the state of a route monitor as UP irrespective of the whether the route entry is present or not in the internal routing table.	The NetScaler appliance displays the state of the route monitor as DOWN if the corresponding route entry is not present in the internal routing table.

A route monitor starts monitoring its route after 180 seconds in the following cases [This is done to allow dynamic routes to get learnt, which may take 180 secs]: • reboot • failover • set route6 command for v6 routes • set route msr enable/disable command for v4 routes. • adding a new route monitor	-

Route monitors are useful in a non-INC mode HA configuration where you want the non-reachability of a gateway from a primary node to be one of the conditions for HA failover.

Consider an example of a non-Inc mode HA setup in a two-arm topology that has NetScaler appliances NS1 and NS2 in the same subnet, with router R1 and switches SW1, SW2, and SW3.

Because R1 is the only router in this setup, you want the HA setup to failover whenever R1 is not reachable from the current primary node. You can configure a route monitor (say, RM1 and RM2, respectively) on each of the nodes to monitor the reachability of R1 from that node.

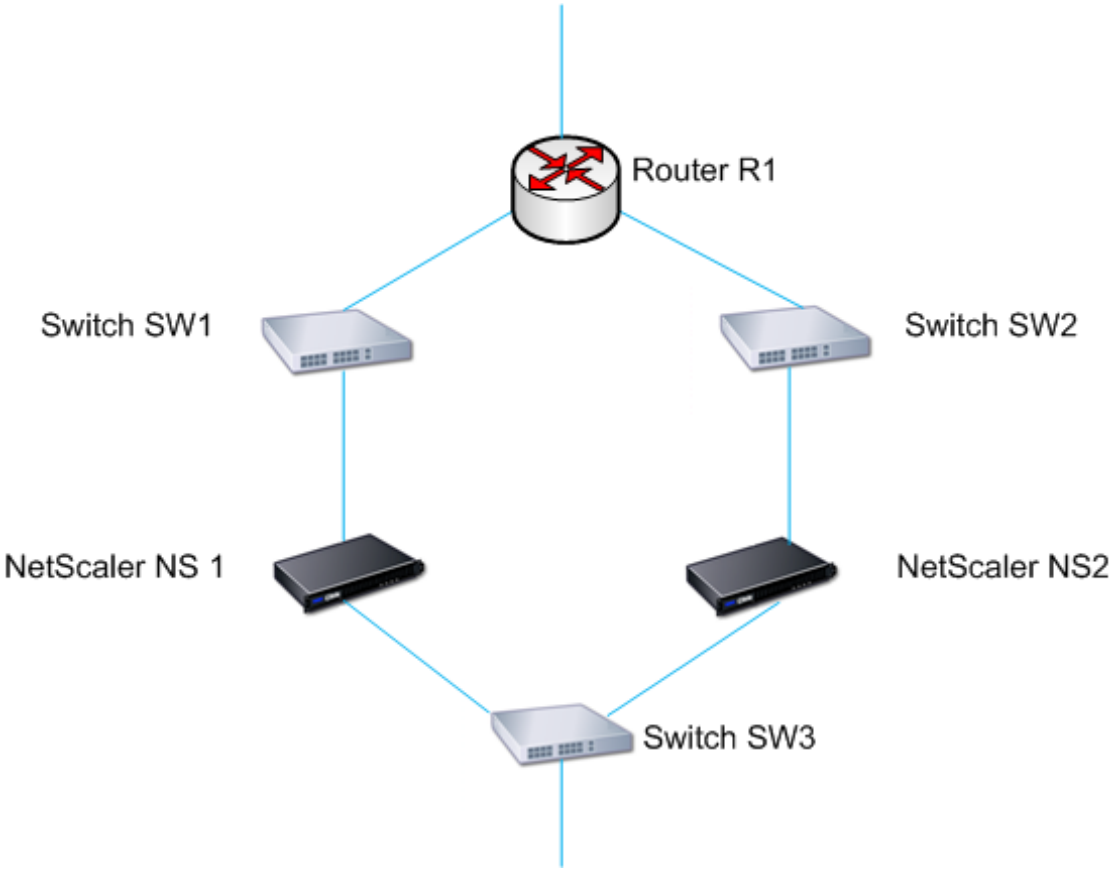


Figure 1.

With NS1 as the current primary node, the execution flow is as follows:

1. Route monitor RM1 on NS1 monitors NS1's internal routing table for the presence of a route entry for router R1. NS1 and NS2 exchange heartbeat messages through switch SW1 or SW3 at regular intervals.

2. If switch SW1 goes down, the routing protocol on NS1 detects that R1 is not reachable and therefore removes the route entry for R1 from the internal routing table. NS1 and NS2 exchanges heartbeat messages through switch SW3 at regular intervals.

3. Detecting that the route entry for R1 is not present in the internal routing table, RM1 initiates a failover. If route to R1 is down from both NS1 and NS2, failover happens every 180 seconds till one of the appliances is able to reach R1 and restore the connectivity.

Adding a Route Monitor to a High Availability Node

A single procedure creates a route monitor and binds it to an HA node.

To add a route monitor by using the NetScaler command line

At the NetScaler command prompt, type:

- bind HA node <id> (**-routeMonitor** <ip_addr|ipv6_addr> [<netmask>])
- sh HA node

Example

bind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
bind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b

Parameters for adding a route monitor

id

The ID of the node to which the monitor is to be bound.

routeMonitor

IPv4 or IPv6 address of the route to be monitored.

netmask

Subnet mask for the IPv4 address.

To add a route monitor by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Route Monitors** tab, click **Configure.**
3. In **Bind / Unbind Route Monitor(s)** dialog box, in the **Network** text box, do one of the following:
   - For a IPv4 network, type an IPv4 network address (for example, 10.102.29.30) and in the **Netmask** text box, type a subnet mask (for example, 255.255.255.0).
   - For a IPv6 network, select the **IPv6** check box and type a IPv6 network address (for example, 1000:0000:0000:0000:0005:0600:700a:888b).
4. Click **Add**. The Route Monitor is added and appears in the **Configured Route Monitors** table.
5. Click **OK.**

Removing Route Monitors

To remove a route monitor by using the NetScaler command line

At the NetScaler command prompt, type:

- unbind HA node <id> (**-routeMonitor** <ip_addr|ipv6_addr> [<netmask>])
- sh HA node

Example

unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b

To remove a route monitor by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Route Monitors** tab, click **Configure.**
3. In the **Bind / Unbind Route Monitor(s)** dialog box, under **Configured Route Monitors**, select a route monitor to remove and click **Remove.**
4. Click **OK.**

Configuring FIS

Link redundancy is a way to prevent failover by grouping interfaces so that, when one interface fails, other functioning interfaces are still available. The link redundancy feature allows you to group the two interfaces into a failover interface set (FIS), which prevents the failure of a single link from causing failover to the secondary system unless all of the interfaces on the primary system are nonfunctional.

Each interface in an FIS maintains independent bridge entries. HA MON interfaces that are not bound to an FIS are known as critical interfaces (CI) because if any of them fails, failover is triggered.

Creating or Modifying an FIS

To add an FIS and bind interfaces to it by by using the NetScaler command line

At the NetScaler command prompt, type:

- add fis <name>
- bind fis <name> <ifnum> ...
- sh fis <name>

Example

```
> add fis fis1
 Done
> bind fis fis1 1/3 1/5
 Done
> show fis fis1
1)      FIS: fis1
       Member Interfaces : 1/3 1/5
 Done
```

An unbound interface becomes a critical interface (CI) if it is enabled and HA MON is on.

To unbind an interface from an FIS by using the NetScaler command line

At the NetScaler command prompt, type:

- unbind fis <name> <ifnum> ...
- sh fis <name>

Example

```
> unbind fis fis1 1/3
 Done
> show FIS fis1
```

```
1)      FIS: fis1
        Member Interfaces : 1/5
 Done
```

Parameters for configuring an FIS

name

Name of the FIS to which interfaces are to be bound.

ifnum

Interface number (slot/port notation) to be bound to the FIS.

To configure an FIS by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Failover Interface Set** tab, do one of the following:
   - To create a new FIS, click **Add.**
   - To modify an existing FIS, click **Open.**
3. In the **Create FIS** or **Configure FIS** dialog box, in the **Name** text box, type the name of the FIS.
4. Select an available interface and click **Add** to bind it to the FIS. Repeat to bind additional interfaces.
5. Click **OK**. A message appears in the status bar, stating that the FIS has been configured successfully.

Removing an FIS

When the FIS is removed, its interfaces are marked as critical interfaces.

To remove an FIS by using the NetScaler command line

At the NetScaler command prompt, type:

rm fis <name>

Example

```
> rm fis fis1
  Done
```

To remove an FIS by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Failover Interface Set** tab, select the FIS that you want to remove and click **Remove**.
3. In the **Remove** dialog box, click **Yes.**

Forcing a Node to Fail Over

You might want to force a failover if, for example, you need to replace or upgrade the primary node. You can force failover from either the primary or the secondary node. A forced failover is not propagated or synchronized. To view the synchronization status after a forced failover, you can view the status of the node.

A forced failover fails in any of the following circumstances:

- You force failover on a standalone system.
- The secondary node is disabled.
- The secondary node is configured to remain secondary.

The NetScaler appliance displays a warning message if it detects a potential issue when you run the force failover command. The message includes the information that triggered the warning, and requests confirmation before proceeding.

Forcing Failover on the Primary Node

If you force failover on the primary node, the primary becomes the secondary and the secondary becomes the primary. Forced failover is possible only when the primary node can determine that the secondary node is UP.

If the secondary node is DOWN, the force failover command returns the following error message: "Operation not possible due to invalid peer state. Rectify and retry."

If the secondary system is in the claiming state or inactive, it returns the following error message: "Operation not possible now. Please wait for system to stabilize before retrying."

To force failover on the primary node by using the NetScaler command line

At the NetScaler command prompt, type:

force HA failover

To force failover on the primary node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, click **Force Failover.**
3. In the **Warning** dialog box, click **Yes.**

Forcing Failover on the Secondary Node

If you run the force failover command from the secondary node, the secondary node becomes primary and the primary node becomes secondary. A force failover can occur only if the secondary node's health is good and it is not configured to stay secondary.

If the secondary node cannot become the primary node, or if secondary node was configured to stay secondary (using the STAYSECONDARY option), the node displays the following error message: "Operation not possible as my state is invalid. View the node for more information."

To force failover on the secondary node by using the NetScaler command line

At the NetScaler command prompt, type:

force HA failover

To force failover on the secondary node by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, click **Force Failover.**
3. In the **Warning** dialog box, click **Yes.**

Forcing Failover When Nodes Are in Listen Mode

When the two nodes of an HA pair are running different versions of the system software, the node running the higher version switches to the listen mode. In this mode, neither command propagation nor synchronization works.

Before upgrading the system software on both nodes, you should test the new version on one of the nodes. To do this, you need to force a failover on the system that has already been upgraded. The upgraded system then takes over as the primary node, but neither command propagation or synchronization occurs. Also, all connections need to be re-established.

To force failover when nodes are in listen mode by using the NetScaler command line

At the NetScaler command prompt, type:

force HA failover

To force failover when nodes are in listen mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, click **Force Failover.**
3. In the **Warning** dialog box, click **Yes.**

Forcing the Secondary Node to Stay Secondary

In an HA setup, the secondary node can be forced to stay secondary regardless of the state of the primary node.

For example, suppose the primary node needs to be upgraded and the process will take a few seconds. During the upgrade, the primary node may go down for a few seconds, but you do not want the secondary node to take over; you want it to remain the secondary node even if it detects a failure in the primary node.

When you force the secondary node to stay secondary, it will remain secondary even if the primary node goes down. Also, when you force the status of a node in an HA pair to stay secondary, it does not participate in HA state machine transitions. The status of the node is displayed as STAYSECONDARY.

Forcing the node to stay secondary works on both standalone and secondary nodes. On a standalone node, you must use this option before you can add a node to create an HA pair. When you add the new node, the existing node continues to function as the primary node, and the new node becomes the secondary node.

Note: When you force a system to remain secondary, the forcing process is not propagated or synchronized. It affects only the node on which you run the command.

To force the secondary node to stay secondary by using the NetScaler command line

At the NetScaler command prompt, type:

set node -**hastatus** STAYSECONDARY

To force the secondary node to stay secondary by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open.**
3. In the **Configure Node** dialog box, under **High Availability Status**, select **STAY SECONDARY**.
4. Click **OK.**

Forcing the Primary Node to Stay Primary

In an HA setup, you can force the primary node to remain primary even after a failover. You can enable this option either on a primary node in an HA pair or on a standalone system.

On a standalone system, you must run this command before you can add a node to create an HA pair. When you add the new node, it becomes the primary node. The existing node stops processing traffic and becomes the secondary node in the HA pair.

To force the primary node to stay primary by using the NetScaler command line

At the NetScaler command prompt, type:

set node -**hastatus** STAYPRIMARY

To force the primary node to stay primary by using the configuration utility

1. In the navigation pane, expand **System**, and then click **High Availability**.
2. In the details pane, on the **Nodes** tab, select the local node, and then click **Open.**
3. In the **Configure Node** dialog box, under **High Availability Status**, select **STAY PRIMARY**.
4. Click **OK.**

Understanding the High Availability Health Check Computation

The following table summarizes the factors examined in a health check computation:

- State of the CIs
- State of the FISs
- State of the route monitors

The following table summarizes the health check computation.

Table 1. High Availability Health Check Computation

FIS	CI	Route monitor	Condition
N	Y	N	If the system has any CIs, all of those CIs must be UP.
Y	Y	N	If the system has any FISs, all of those FISs must be UP.
Y	Y	Y	If the system has any route monitors configured, all monitored routes must be present in the FIS.

Troubleshooting High Availability Issues

Certain conditions can cause improper synchronization between nodes or incorrect configuration on the secondary node.

- **Improper synchronization of VLAN configuration in high availability systems**. In HA pairs, synchronization does not work properly if only one node has a VLAN configured. To prevent this problem, configure your VLANs after you configure your appliances as an HA pair, and be sure to configure them both.

- **Retrieving a lost configuration**. If the primary node is unable to send the configuration to the secondary node due to a network error, the secondary node may not have an accurate configuration and may not behave correctly if a failover occurs. If this happens, you can retrieve the current configuration from the configuration backup on the hard disk of the primary appliance. The operating system saves the last four copies of the ns.conf file in the /nsconfig directory as ns.conf.0, ns.conf.1, ns.conf.2, and ns.conf.3. The ns.conf.0 file contains the current configuration.

To retrieve the current system configuration

1. Exit the CLI to FreeBSD by typing the following command and pressing the Enter key:

   > shell

   The FreeBSD shell prompt appears, as shown below.

   #

2. Copy the latest backup file to /nsconfig/ns.conf by using the following command:

   # cp ‘ls -t /nsconfig/ns.conf.? | head -1` /nsconfig/ns.conf

If you perform a configuration using the NSConfig utility, it is not propagated. If you create a configuration using NSconfig, you must repeat the configuration steps separately for each node in an HA pair.

AAA Application Traffic

Many companies restrict web site access to valid users only, and control the level of access permitted to each user. The AAA feature allows a site administrator to manage access controls with the NetScaler appliance instead of managing these controls separately for each application. Doing authentication on the appliance also permits sharing this information across all web sites within the same domain that are protected by the appliance.

The AAA feature supports authentication, authorization, and auditing for all application traffic. To use AAA, you must configure authentication virtual servers to handle the authentication process and traffic management virtual servers to handle the traffic to web applications that require authentication. You also configure your DNS to assign FQDNs to each virtual server. After configuring the virtual servers, you configure a user account for each user that will authenticate via the NetScaler appliance, and optionally you create groups and assign user accounts to groups. After creating user accounts and groups, you configure policies that tell the appliance how to authenticate users, which resources to allow users to access, and how to log user sessions. To put the policies into effect, you bind each policy globally, to a specific virtual server, or to the appropriate user accounts or groups. After configuring your policies, you customize user sessions by configuring session settings and binding your session policies to the traffic management virtual server. Finally, if your intranet uses client certs, you set up the client certificate configuration.

Before configuring AAA, you should be familiar with and understand how to configure load balancing, content switching, and SSL on the NetScaler appliance. For more information about load balancing, content switching, and SSL, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

How AAA Works

AAA provides security for a distributed Internet environment by allowing any client with the proper credentials to connect securely to protected application servers from anywhere on the Internet. This feature incorporates the three security features of authentication, authorization, and auditing. Authentication enables the NetScaler appliance to verify the client's credentials, either locally or with a third-party authentication server, and allow only approved users to access protected servers. Authorization enables the appliance to verify which content on a protected server it should allow each user to access. Auditing enables the appliance to keep a record of each user's activity on a protected server.

To understand how AAA works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the NetScaler appliance does the following:

- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through LDAP.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.
- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

Authentication requires that several entities—the client, the NetScaler appliance, the external authentication server if one is used, and the application server—respond to each other when prompted by performing a complex series of tasks in the correct order. If you are using an external authentication server, this process can be broken down into the following fifteen steps.

- The client sends a GET request for a URL on the application server.
- The NetScaler appliance's traffic management virtual server redirects the request to the application server.
- The application server determines that the client has not been authenticated, and therefore sends an HTTP 200 OK response via the TM vserver to the client. The response contains a hidden script that causes the client to issue a POST request for /cgi/tm.
- The client sends a POST request for /cgi/tm.
- The NetScaler appliance's authentication virtual server redirects the request to the authentication server.

- The authentication server creates an authentication session, sets and caches a cookie that consists of the initial URL and the domain of the traffic management virtual server, and then sends an HTTP 302 response via the authentication virtual server, redirecting the client to /vpn/index.html.
- The client sends a GET request for /vpn/index.html.
- The authentication virtual server redirects the client to the authentication server login page.
- The client sends a GET request for the login page, enters credentials, and then sends a POST request with the credentials back to the login page.
- The authentication virtual server redirects the POST request to the authentication server.
- If the credentials are correct, the authentication server tells the authentication virtual server to log the client in and redirect the client to the URL that was in the initial GET request.
- The authentication virtual server logs the client in and sends an HTTP 302 response that redirects the client to the initially requested URL.
- The client sends a GET request for their initial URL.
- The traffic management virtual server redirects the GET request to the application server.
- The application server responds via the traffic management virtual server with the initial URL.

If you use local authentication, the process is similar, but the authentication virtual server handles all authentication tasks instead of forwarding connections to an external authentication server. The following figure illustrates the authentication process.

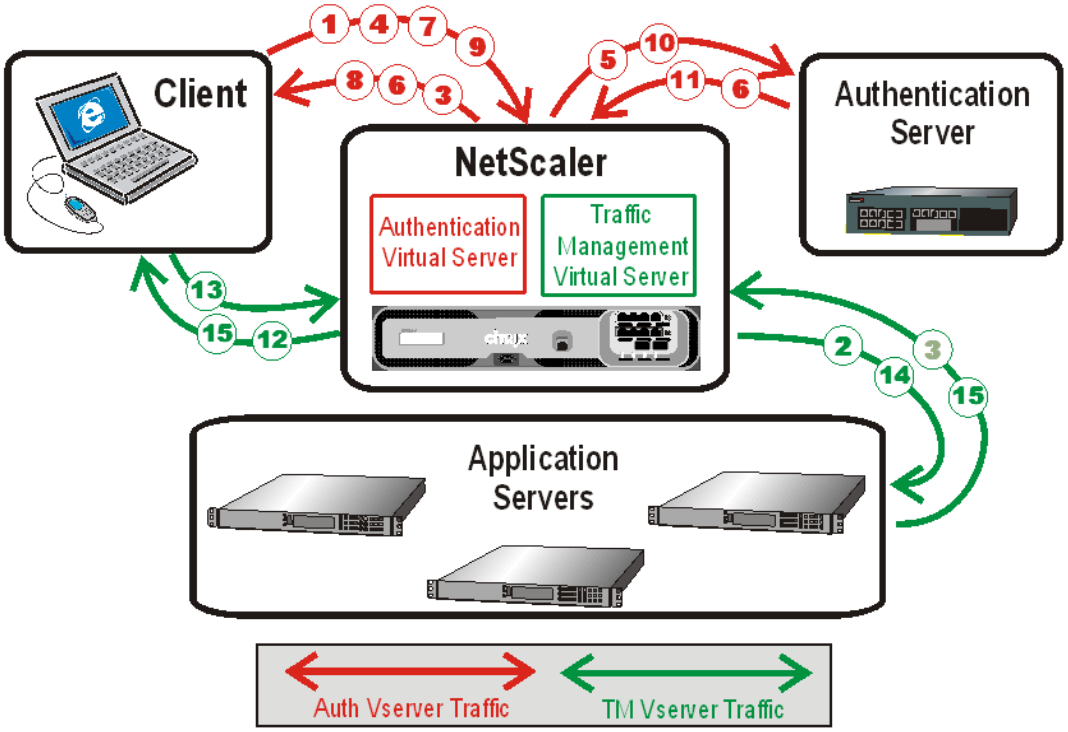


Figure 1. Authentication Process Traffic Flow

When an authenticated client requests a resource, the NetScaler appliance, before sending the request to the application server, checks the user and group policies associated with the client account, to verify that the client is authorized to access that resource. The appliance handles all authorization on protected application servers. You do not need to do any special configuration of your protected application servers.

The NetScaler appliance supports auditing of all states and status information, so you can see the details of what each user did while logged on, in chronological order. To provide this information, the appliance logs each event, as it occurs, either to a designated audit log file on the appliance or to a syslog server. Auditing requires configuring the appliance and any syslog server that you use.

How AAA Works

AAA provides security for a distributed Internet environment by allowing any client with the proper credentials to connect securely to protected application servers from anywhere on the Internet. This feature incorporates the three security features of authentication, authorization, and auditing. Authentication enables the NetScaler appliance to verify the client's credentials, either locally or with a third-party authentication server, and allow only approved users to access protected servers. Authorization enables the appliance to verify which content on a protected server it should allow each user to access. Auditing enables the appliance to keep a record of each user's activity on a protected server.

To understand how AAA works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the NetScaler appliance does the following:

- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through LDAP.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.
- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

Authentication requires that several entities—the client, the NetScaler appliance, the external authentication server if one is used, and the application server—respond to each other when prompted by performing a complex series of tasks in the correct order. If you are using an external authentication server, this process can be broken down into the following fifteen steps.

- The client sends a GET request for a URL on the application server.
- The NetScaler appliance's traffic management virtual server redirects the request to the application server.
- The application server determines that the client has not been authenticated, and therefore sends an HTTP 200 OK response via the TM vserver to the client. The response contains a hidden script that causes the client to issue a POST request for /cgi/tm.
- The client sends a POST request for /cgi/tm.
- The NetScaler appliance's authentication virtual server redirects the request to the authentication server.

- The authentication server creates an authentication session, sets and caches a cookie that consists of the initial URL and the domain of the traffic management virtual server, and then sends an HTTP 302 response via the authentication virtual server, redirecting the client to /vpn/index.html.
- The client sends a GET request for /vpn/index.html.
- The authentication virtual server redirects the client to the authentication server login page.
- The client sends a GET request for the login page, enters credentials, and then sends a POST request with the credentials back to the login page.
- The authentication virtual server redirects the POST request to the authentication server.
- If the credentials are correct, the authentication server tells the authentication virtual server to log the client in and redirect the client to the URL that was in the initial GET request.
- The authentication virtual server logs the client in and sends an HTTP 302 response that redirects the client to the initially requested URL.
- The client sends a GET request for their initial URL.
- The traffic management virtual server redirects the GET request to the application server.
- The application server responds via the traffic management virtual server with the initial URL.

If you use local authentication, the process is similar, but the authentication virtual server handles all authentication tasks instead of forwarding connections to an external authentication server. The following figure illustrates the authentication process.

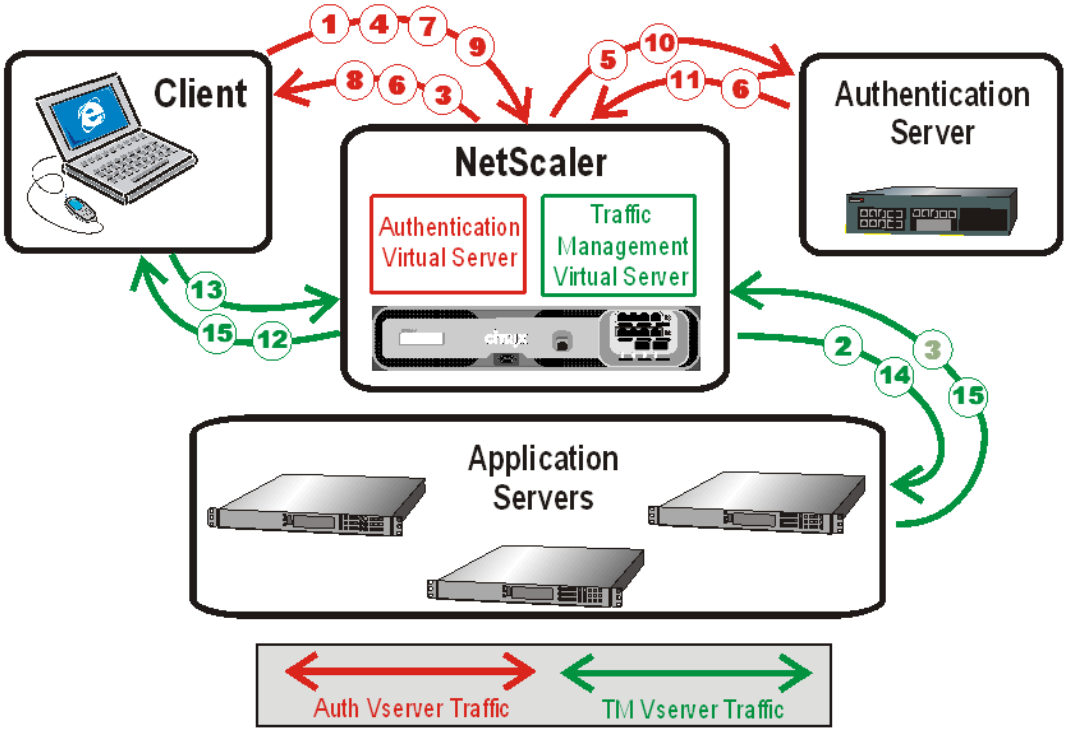


Figure 1. Authentication Process Traffic Flow

When an authenticated client requests a resource, the NetScaler appliance, before sending the request to the application server, checks the user and group policies associated with the client account, to verify that the client is authorized to access that resource. The appliance handles all authorization on protected application servers. You do not need to do any special configuration of your protected application servers.

The NetScaler appliance supports auditing of all states and status information, so you can see the details of what each user did while logged on, in chronological order. To provide this information, the appliance logs each event, as it occurs, either to a designated audit log file on the appliance or to a syslog server. Auditing requires configuring the appliance and any syslog server that you use.

Enabling AAA

To use the AAA feature, you must enable it. You can configure AAA entities—such as the authentication and traffic management virtual servers—before you enable the AAA feature, but the entities will not function until the feature is enabled.

To enable AAA by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable AAA and verify the configuration:

- enable feature AAA
- show ns feature

Example

```
> enable feature AAA
 Done

> show ns feature

      Feature                 Acronym           Status
      -------                 -------           ------
 1)   Web Logging               WL                OFF
 2)   Surge Protection          SP                ON
 .
 .
 .
 15)  AAA                     AAA               ON
 .
 .
 .
 23)  HTML Injection          HTMLInjection     ON
 24)  NetScaler Push          push              OFF
Done
```

To enable AAA by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Authentication, Authorization and Auditing** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)?** message box, click **Yes**. A message appears in the status bar, stating that the feature has been enabled.

Setting up AAA Virtual Servers and DNS

You can configure AAA by using the built-in wizard, or manually. To use the wizard, in the main AAA pane of the configuration utility, you click **AAA - Application Traffic wizard** and follow the prompts.

To configure AAA manually, you first configure an authentication virtual server, which involves binding an SSL certificate-key pair. You then associate the authentication virtual server with a new or existing traffic management virtual server. (Either a load balancing virtual server or a content switching virtual server can serve as a traffic management virtual server.) To complete the initial configuration, you configure DNS to assign hostnames to both the authentication virtual server and the traffic management virtual server, and verify that your virtual servers are UP and configured correctly.

Caution: Both virtual servers must have hostnames in the same domain, or the AAA configuration will not work.

Configuring the Authentication Virtual Server

To configure AAA, you first configure an authentication virtual server to handle authentication traffic. You must bind an SSL certificate-key pair to the virtual server to enable it to handle SSL connections. For additional information about configuring SSL and creating a certificate-key pair, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To configure an authentication virtual server by using the NetScaler command line

At a NetScaler command prompt, to configure an authentication virtual server and verify the configuration, type the following commands in the order shown:

- add authentication vserver <name> ssl <ipaddress>
- show authentication vserver <name>
- bind ssl certkey <sslkeyname> <name>
- show authentication vserver <name>
- set authentication vserver <name> **-authenticationDomain** <FQDN>
- show authentication vserver <name>

Example

```
> add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443
 Done
> show authentication vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: DOWN[Certkey not bound]
      Client Idle Timeout: 180 sec
      Down state flush: DISABLED
      Disable Primary Vserver On Down : DISABLED
      Authentication : ON
      Current AAA Users: 0
 Done
> bind ssl certkey Auth-Vserver-2 Auth-Cert-1
 Done
> show authentication vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: UP
```

```
        Client Idle Timeout: 180 sec
        Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
        Authentication Domain: myCompany.employee.com
 Done
> set authentication vserver Auth-Vserver-2 -AuthenticationDomain myCompany.employee.com
 Done
> show authentication vserver Auth-Vserver-2
        Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
        State: DOWN[Certkey not bound]
        Client Idle Timeout: 180 sec
        Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
        Authentication Domain: myCompany.employee.com
 Done
```

Parameters for configuring the authentication virtual server

name

A name for your new authentication virtual server, or the name of an existing authentication virtual server. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed after the virtual server is created.)

ipaddress

IP address assigned to the authentication virtual server in DNS.

sslkeyname

Name of the SSL certificate-key pair to associate with the virtual server.

authenticationDomain

The fully qualified domain name to be assigned to the authentication virtual server in your DNS. Must match the domain name in the SSL certificate-key pair.

To configure an authentication virtual server by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic** and click **Virtual Servers**.
2. In the details pane, do one of the following:
   - To create a new authentication virtual server, click **Add**.
   - To modify an existing authentication virtual server, select the virtual server, and then click **Open**.
3. In the **Create Virtual Server (Authentication) or Configure Virtual Server (Authentication)** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the authentication virtual server" as follows (asterisk indicates a required parameter):
   - Name*—name (Cannot be changed for a previously created virtual server)
   - IP Address*—ipaddress
   - Domain*—authenticationDomain

   Note: The authentication virtual server uses only the SSL protocol and port 443, so those options are greyed out.
4. On the **Certificates** tab, in the **Available** list, select the SSL certificate you want to associate with this authentication virtual server, and click the **Add** button. If your configuration requires CA certs and you will use this SSL certificate as the CA server certificate, you click the **Add as CA** button instead. The certificate moves from the **Available** to the **Configured** list.
5. Click **Create** or **OK**, and then click **Close**. If you created a new authentication virtual server, it now appears in the **Authentication Virtual Servers** pane.

Configuring a Traffic Management Virtual Server

After you have created and configured your authentication virtual server, you next create or configure a traffic management virtual server and associate your authentication virtual sever with it. You can use either a load balancing or content switching virtual server for a traffic management virtual server. For more information about creating and configuring either type of virtual server, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Note: The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

You configure a traffic management virtual server for AAA by enabling authentication and then assigning the FQDN of the authentication server to the traffic management virtual server. You can also configure the authentication domain on the traffic management virtual server at this time. If you do not configure this option, the NetScaler appliance assigns the traffic management virtual server an FQDN that consists of the FQDN of the authentication virtual server without the hostname portion. For example, if domain name of the authentication vserver is `tm.xyz.bar.com`, the appliance assigns `xyz.bar.com`. as the authentication domain.

To configure a TM virtual server for AAA by using the NetScaler command line

At the NetScaler command prompt, type one of the following sets of commands to configure a TM virtual server and verify the configuration:

- set lb vserver <name> **-authentication** ON **-authenticationhost** <FQDN> [**-authenticationdomain** <authdomain>]
- show lb vserver <name>
- set cs vserver <name> **-authentication** ON **-authenticationhost** <FQDN> [**-authenticationdomain** <authdomain>]
- show cs vserver <name>

Example

```
> set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki.index.com
 Done
```

```
> show lb vserver vs-cont-sw
        vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:15 2009 (+410 ms)
        Time since last state change: 5 days, 20:00:40.290
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
        Authentication: ON      Host: mywiki.index.com
 Done
```

Parameters for configuring a traffic management virtual server

name

The name of the load balancing or content switching virtual server being configured as a traffic management virtual server.

authentication

Toggle authentication of application traffic on the traffic management virtual server. Possible values: ON, OFF. Default: OFF.

authenticationhost

FQDN assigned to the authentication virtual server.

authenticationdomain

The common domain in the FQDNs of both the authentication virtual server and the traffic management virtual server.

To configure a TM virtual server for AAA by using the configuration utility

1. In the navigation pane, do one of the following.
   - Expand **Load Balancing**, and then click **Virtual Servers**.
   - Expand **Content Switching**, and then click **Virtual Servers**.

   The AAA configuration process for either type of virtual server is identical.
2. In the details pane, select the virtual server on which you want to enable authentication, and then click **Open**.
3. In the **Domain** text box, type the authentication domain. See `authenticationdomain`, in "Parameters for configuring a traffic management virtual server," for information about this parameter.
4. On the **Advanced** tab, select the **Authentication** check box.
5. In the **Authentication Host** text box, type the fully qualified domain name of the authentication virtual server. See authenticationhost in the table above for information about this parameter.
6. Click **OK**. A message appears in the status bar, stating that the vserver has been configured successfully.

Configuring DNS

For the domain session cookie used in the authentication process to function correctly, you must configure DNS to assign both the authentication and the traffic management virtual servers to FQDNs in the same domain. For information about how to the configure DNS address records, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Verifying Your Setup for AAA

After you configure authentication and traffic management virtual servers and before you create user accounts, you should verify that both virtual servers are configured correctly and are in the UP state. For information about verifying the setup on your traffic management server, see the appropriate chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

To verify authentication virtual server setup by using the NetScaler command line

At the NetScaler command prompt, type the following command:

show authentication vserver <name>

Example

```
> show authentication vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: UP
      Client Idle Timeout: 180 sec
      Down state flush: DISABLED
      Disable Primary Vserver On Down : DISABLED
      Authentication : ON
      Current AAA Users: 0
      Authentication Domain: myCompany.employee.com
 Done
```

Parameters for Verifying your Setup for AAA

State

Current state of the service or virtual server. Possible values:

UP – The virtual server can respond to requests. Your authentication virtual server should be UP.

OUT OF SERVICE – The virtual server has been manually disabled. New requests received by this virtual server are dropped unless a backup virtual server or HTTP redirection is configured.

DOWN – The virtual server cannot respond to requests. An authentication virtual server is DOWN when a valid SSL certificate-key pair is not bound to it.

Client Idle Timeout

Idle time (in seconds) after which client connections are terminated. Default value for HTTP/SSL-based services: 180.

Down state flush

Perform delayed cleanup of connections on this virtual server. Possible values: ENABLED, DISABLED. Default: ENABLED.

Disable Primary Vserver On Down

Keep the primary virtual server secondary, when it comes back up, until manually forced to take over as primary. If enabled, preserves database updates on the backup, enabling you to synchronize the databases before restoring the primary. Possible values: ENABLED, DISABLED. Default: DISABLED.

Authentication

Authenticate application traffic for the traffic management virtual server. Possible values: ON, OFF. Default: OFF. This value must be ON for AAA to function.

Current AAA Users

Number of AAA users configured. This number should be zero if you have just started to create a AAA configuration.

Authentication Domain

Authentication domain configured for the authentication virtual server.

Beneath this information are listed any policies bound globally or to this authentication virtual server, and their priorities.

To verify your AAA virtual server setup by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Virtual Servers**.
2. Review the information in the **AAA Virtual Servers** pane to verify that your configuration is correct and your authentication virtual server is accepting traffic. You can select a specific virtual server to view detailed information in the details pane.

   Note: For descriptions of what the information signifies, see the list above.

Configuring Users and Groups

After configuring the AAA basic setup, you create users and groups. You first create a user account for each person who will authenticate via the NetScaler appliance. If you are using local authentication controlled by the NetScaler appliance itself, you create local user accounts and assign passwords to each of those accounts.

You also create user accounts on the NetScaler appliance if you are using an external authentication server. In this case, however, each user account must exactly match an account for that user on the external authentication server, and you do not assign passwords to the user accounts that you create on the NetScaler. The external authentication server manages the passwords for users that authenticate with the external authentication server.

If you are using an external authentication server, you can still create local user accounts on the NetScaler appliance if, for example, you want to allow temporary users (such as visitors) to log in but do not want to create entries for those users on the authentication server. You assign a password to each local user account, just as you would if you were using local authentication for all user accounts.

Each user account must be bound to policies for authentication and authorization. To simplify this task, you can create one or more groups and assign user accounts to them. You can then bind policies to groups instead of individual user accounts.

To create a local AAA user account by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a local AAA user account and verify the configuration:

- add aaa user <username> [-password <password>]
- `show aaa user`

Example

```
> add aaa user user-2 -password emptybag
 Done
> show aaa user
1)      UserName: user-1
2)      UserName: user-2
 Done
```

To change the password for an existing AAA local user account by using the NetScaler command line

At the NetScaler command prompt, type the following command and, when prompted, type the new password:

set aaa user <username>

Example

```
> set aaa user user-2
Enter password:
Done
```

Parameters for Configuring AAA Local Users

username

A name for the user. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for an existing user.)

password

A password that the user uses to log in. This parameter is required for all user accounts if you are not using an external authentication server. If you are using an external authentication server, you provide a password only for local user accounts that do not exist on the authentication server.

To configure AAA local users by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Users**.
2. In the details pane, do one of the following:
   - To create a new user account, click **Add**.
   - To modify an existing user account, select the user account, and then click **Open**.
3. In the **Create AAA User** dialog box, in the **User Name** text box, type a name for the user. For rules for user names, see the list above.
4. If creating a locally authenticated user account, clear the **External Authentication** check box and provide a local password that the user will use to log on.
5. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the user has been configured successfully.

To create AAA local groups and add users to them by using the NetScaler command line

At the NetScaler command prompt, type the following commands. Type the first command one time, and type the second command once for each user:

- add aaa group <groupname>
- show aaa group

  Example

  ```
  > add aaa group group-2
  Done
  > show aaa group
  1)      GroupName: group-1
  2)      GroupName: group-2
  Done
  ```

- bind aaa group <groupname> -username <username>

  Example

  ```
  > bind aaa group group-2 -username user-2
   Done
  > show aaa group group-2
        GroupName: group-2

                UserName: user-2
   Done
  ```

To remove users from an AAA group by using the NetScaler command line

At the NetScaler command prompt, unbind users from the group by typing the following command once for each user account that is bound to the group:

unbind aaa group <groupname> -username <username>

Example

```
> unbind aaa group group-hr -username user-hr-1
 Done
```

To remove an AAA group by using the NetScaler command line

First remove all users from the group. Then, at the NetScaler command prompt, type the following command to remove an AAA group and verify the configuration:

- rm aaa group <groupname>
- show aaa group

Example

```
> rm aaa group group-hr
 Done
> show aaa group
1)      GroupName: group-1
2)      GroupName: group-finance
 Done
```

Parameters for Configuring AAA Local Groups

groupname

A name for the group you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing groups.)

username

The name of a user account to be added to the new group.

To configure AAA local groups and add users to them by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Groups**.
2. In the details pane, do one of the following:
   - To create a new group, click **Add**.
   - To modify an existing group, select the group, and then click **Open**.
3. If you are creating a new group, in the **Create AAA Group** dialog box, in the **Group Name** text box, type a name for the group. For information about group names, see the list above, under groupname.
4. On the **Users** tab, configure the users assigned to the group.
   a. To add a user to the group, in the **Available Users** list, select the user, and then click **Add**.
   b. To remove a user from the group, in the **Configured Users** list, select the user, and then click **Remove**.
   c. To create a new user account and add it to the group, click **New**, and then follow the instructions in “To configure AAA local users by using the configuration utility.”
5. Click **Create** or **OK**, and then click **Close**. The group that you created appears in the **AAA Groups** page.

Configuring AAA Policies

After you set up your users and groups, you next configure authentication policies, authorization policies, and audit policies to define which users are allowed to access your intranet, which resources each user or group is allowed to access, and what level of detail AAA will preserve in the audit logs. An authentication policy defines the type of authentication to apply when a user attempts to log on. If external authentication is used, the policy also specifies the external authentication server. Authorization policies specify the network resources that users and groups can access after they log on. Auditing policies define the audit log type and location.

You must bind each policy to put it into effect. You bind authentication policies to authentication virtual servers, authorization policies to one or more user accounts or groups, and auditing policies both globally and to one or more user accounts or groups.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler operating system, policy priorities work in reverse order: the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The AAA feature implements only the first of each type of policy that a request matches, not any additional policies of that type that a request might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind the policies. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Authentication Policies

The NetScaler appliance can authenticate users with local user accounts or by using an external authentication server. The appliance supports the following authentication types:

LOCAL

Authenticates to the NetScaler by using a password, without reference to an external authentication server. User data is stored locally on the NetScaler appliance.

RADIUS

Authenticate to an external Radius server.

LDAP

Authenticates to an external LDAP authentication server.

TACACS

Authenticates to an external Terminal Access Controller Access-Control System (TACACS) authentication server.

NT4

Authenticates to an external Windows NT 4.0 server.

CERT

Authenticates to the NetScaler by using a client certificate, without reference to an external authentication server.

NEGOTIATE

Authenticates to a Kerberos authentication server. If there is an error in Kerberos authentication, NetScaler uses NTLM authentication. For more information on Kerberos authentication, see Handling Authentication, Authorization and Auditing (AAA) with Kerberos.

An authentication policy is comprised of an expression and an action. Authentication policies use NetScaler expressions, which are described in detail in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

After creating an authentication policy, bind it to an authentication virtual server and assign a priority to it. When binding it, also designate it as either a primary or a secondary policy. Primary policies are evaluated before secondary policies. In configurations that use both types of policy, primary policies are normally more specific policies while secondary policies are normally more general policies intended to handle authentication for any user accounts that do not meet the more specific criteria.

To create and bind an authentication policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands in the order shown to create and bind an authentication policy and verify the configuration:

- add authentication negotiatePolicy <name> <rule> <reqAction>
- show authorization policy <policyname>
- bind authentication vserver <authvsname> -**policy** <policyname> [-**priority** <priority>] [-secondary]]
- show authentication Vserver <name>

Example

```
> add authentication localPolicy Authn-Pol-1 ns_true
 Done

> show authentication localPolicy
1)      Name: Authn-Pol-1       Rule: ns_true
       Request action: LOCAL
 Done
> bind authentication Vserver Auth-Vserver-2 -policy Authn-Pol-1
 Done
> show authentication Vserver Auth-Vserver-2
       Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
       State: UP
       Client Idle Timeout: 180 sec
       Down state flush: DISABLED
       Disable Primary Vserver On Down : DISABLED
       Authentication : ON
       Current AAA Users: 0
       Authentication Domain: myCompany.employee.com

1)            Primary authentication policy name: Authn-Pol-1 Priority: 0
 Done
```

To modify an existing authentication policy

At the NetScaler command prompt, type the following commands to modify an existing authentication policy:

set authentication <authtype> <policyname> <rule> [**-reqaction** <action>]

Example

```
> set authentication localPolicy Authn-Pol-1 'ns_true'
Done
```

To remove an authentication policy

At the NetScaler command prompt, type the following command to remove an authentication policy:

rm authentication <authtype> <policyname>

Example

```
> rm authentication localPolicy Authn-Pol-1
Done
```

Parameters for configuring authentication policies

authType

Type of authentication. Possible values: LOCAL, RADIUS, LDAP, TACACS, NT4, CERT, NEGOTIATE. Default value: LOCAL

policyName

A name for the policy you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing policies.)

rule

An expression that defines the requests to be authenticated. For a complete description of NetScaler expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

reqAction

The action associated with your policy. Leave this blank unless you are using an external authentication server that requires an action.

authVsName

The name of the authentication virtual server to which you are binding this policy.

priority

The priority assigned to this authentication policy.

secondary

Designate this policy as a secondary authentication policy.

To configure and bind authentication policies by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, expand **Policies**, and then click **Authentication**.
2. In the details pane, on the **Policies** tab, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the action, and then click **Open**.
3. In the **Create Authentication Policy** or **Configure Authentication Policy** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring authentication policies" as follows (asterisk indicates a required parameter):
   - **Name***—policyname(Cannot be changed for a previously configured action)
   - **Authentication Type***—authtype
   - **Server***—action
   - **Expression***—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the **Expression** window, and then by typing your expression directly into the expression text area, or by clicking **Add** to open **Add Expression** dialog box and using the drop-down lists in it to construct your expression.)
4. Click **Create** or **OK**, and then click **Close**. The policy that you created appears in the **Authentication Policies and Servers** page.
5. Click the **Servers** tab.
6. In the details pane, select the appropriate virtual server, and then click **Open**.
7. If you want to designate this policy as a secondary authentication policy, on the **Authentication** tab, click **Secondary**. If you want to designate this policy as a primary authentication policy, skip this step.
8. Click **Insert Policy**.
9. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
10. In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
11. Click **OK**. A message appears in the status bar, stating that the policy has been configured successfully.

Authorization Policies

After you create authentication policies, you next create any authorization policies you need. Authorization policies, like other policies, consist of an expression and action. There are only two actions for authorization policies: ALLOW and DENY. ALLOW permits users to access the specified resource; DENY blocks access. The default setting for authorization when no specific policy exists is to deny access to network resources. This means that a user or group can access a particular resource only if an authorization policy explicitly allows access. For optimum security, the best practice is not to change the default setting and to create specific authorization policies for users who need access to specific resources.

Authorization use both default syntax expressions and classic expressions. These expressions are described in detail in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

After you create an authorization policy, you bind it to the appropriate user accounts or groups to put it into effect.

To create an authorization policy

At the NetScaler command prompt, type the following commands to create an authorization policy and verify the configuration:

- add authorization policy <policyname> <rule> **-action** <action>
- show authorization policy <name>

Example

```
> add authorization policy authz-pol-1 "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" DENY
 Done
> show authorization policy authz-pol-1
1)      Name: authz-pol-1    Rule: HTTP.REQ.URL.SUFFIX.EQ("gif")
       Action: DENY
 Done
>
```

For the classic syntax equivalent of the above example, see To create an authorization policy.

To modify an authorization policy

At the NetScaler command prompt, type the following command to modify an authorization policy:

set authorization policy <policyname> [**-rule** <expression>] **-action** <action>

Example

```
> set authorization policy authz-pol-1 -rule "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" -action ALLOW
 Done
> show authorization policy authz-pol-1
1)      Name: authz-pol-1    Rule: HTTP.REQ.URL.SUFFIX.EQ("gif")
       Action: ALLOW
 Done
>
```

For the classic syntax equivalent of the above example, see To modify an authorization policy.

To bind an authorization policy to a user account or group

At the NetScaler command prompt, type one of the following commands to bind an authorization policy to a user account or group and verify the configuration:

- bind aaa user <userName> [**-policy** <policyname> [**-priority** <priority]] [**-intranetApplication** <appname>] [**-urlName** <urlname>] [**-intranetIP** <intranetip> [<netmask>]]
- show aaa user <name>
- bind aaa group <groupName> [**-policy** <policyname> [**-priority** <priority]] [**-intranetApplication** <appname>] [**-urlName** <urlname>] [**-intranetIP** <intranetip> [<netmask>]]
- show aaa group <name>

Example

```
> bind aaa user user-hr-1 -policy authz-pol-1
  Done
> show aaa user user-hr-1
       UserName: user-hr-1

              Policy: authz-pol-1, Priority: 0
```

```
 Done
> bind aaa group group-1 -policy authz-pol-1
 Done
> show aaa group group-1
      GroupName: group-1

            UserName: user-2
            UserName: user-1

            Policy: authz-pol-1, Priority: 0
 Done
```

To unbind an authorization policy from a user account or group

At the NetScaler command prompt, type one of the following commands to unbind an authorization policy from a user account or group:

- unbind aaa user <userName> **-policy** <policyname>
- unbind aaa group <groupName> **-policy** <policyname>

Example

```
> unbind aaa user aaa-user-1 -policy auth-pol-1
Done
```

To remove an authorization policy

First unbind the policy from all user accounts and groups, and then, at the NetScaler command prompt, type the following command to remove an authorization policy:

rm authorization policy <policyname>

Parameters for configuring authorization policies

policyname

A name for the authorization policy you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing policies.)

rule

A NetScaler default syntax or classic syntax expression that defines which requests to allow or deny. For a complete description of NetScaler default syntax and classic syntax expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

action

The action to perform when a connection matches the policy. Possible values: ALLOW, DENY. Default: ALLOW.

username or groupname

The name of the user account or the group to which you are binding the authorization policy.

priority

The priority you are assigning to the policy.

internetApplication

The name of the intranet application to which you are binding the authorization policy.

urlname

The URL of the intranet application to which you are binding the authorization policy.

intranetip

The intranet IP of the intranet application to which you are binding the authorization policy.

netmask

If the intranet application to which you are binding the authorization policy resides on an IP range, the subnet mask of that intranet range.

To configure and bind authorization policies by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Authorization**.

2. In the details pane, do one of the following:

   - To create a new authorization policy, click **Add**.

   - To modify an existing authorization policy, select the policy, and then click **Open**.

3. In the **Create Authorization Policy** or **Configure Authorization Policy** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring authorization policies" as follows (asterisk indicates a required parameter):

   - Name*—policyname(Cannot be changed for a previously configured policy.)

   - Action*— action

   - Expression*—rule (By default, the **Expression** box accepts default syntax policies. To switch to the classic syntax view, click **Switch to Classic Syntax**.)

4. Click **Create** or **OK**, and then click **Close**. The policy that you created appears on the **Authorization Policies** page.

5. To bind an authorization policy to a user account or group, in the navigation pane, under **AAA - Application Traffic**, click either **Users** or **Groups**, as appropriate, and then add that policy to the user account list:

   a. In the details pane, select the appropriate user account, and then click **Open**.

   b. Click the **Authorization** tab.

   c. Click **Insert Policy**.

   d. Choose the policy you want to bind to the group from the drop-down list.

   e. In the **Priority** column, modify the default priority as needed to ensure that the policy is evaluated in the proper order.

   f. Click **OK**.

   A message appears in the status bar, stating that the policy has been configured successfully.

Auditing Policies

After you create authentication policies, you next create any auditing policies you need. The NetScaler appliance allows auditing of all states and status information, so you can see the event history for any user in chronological order. When you configure auditing on the NetScaler appliance, you can choose to store the log files locally on the NetScaler appliance or to send them to a syslog server.

To put your auditing policies into effect, you bind them globally, to a specific authentication virtual server, or to specific user accounts or groups.

To create an auditing policy by using the NetScaler command line

At the NetScaler command prompt, type the following commnads to create an auditing policy and verify the configuration:

- add audit <logtype> <policyname> [**-rule** <rule>] [**-action** <action>]
- show audit nslogPolicy

Example

```
> add audit nslogPolicy audit-1 ns_true audit_server
 Done
> show audit nslogPolicy
1)      Name: audit-pol Rule: ns_true
       Action: audit_server
2)      Name: audit-1   Rule: ns_true
       Action: audit_server
 Done
```

To modify an auditing policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify an auditing policy and verify the configuration:

- set audit <logtype> <policyname> **-rule** [<rule>] [**-action** <action>]
- show audit nslogPolicy

Example

```
> set audit nslogPolicy audit-1 ns_true audit_server
 Done
> show audit nslogPolicy
1)      Name: audit-pol Rule: ns_true
       Action: audit_server
2)      Name: audit-1   Rule: ns_true
       Action: audit_server
 Done
```

To globally bind an auditing policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind an auditing policy:

bind audit <logtype> global **-policy** <policyname> [**-priority** <priority]

Example

```
> bind tm global -policyName Audit-Pol-1 -priority 1000
Done
```

To bind an auditing policy to an authentication virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an auditing policy to an authentication virtual server and verify the configuration:

- bind authentication vserver <authvsname> **-policy** <policyname> [**-priority** <priority>]
- show authentication verserver <name>

Example

```
> bind authentication Vserver Auth-Vserver-2 -policy Authn-Pol-1
 Done
> show authentication Vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: UP
      Client Idle Timeout: 180 sec
```

```
        Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
        Authentication Domain: myCompany.employee.com

1)          Primary authentication policy name: Authn-Pol-1 Priority: 0
 Done
```

To bind an auditing policy to a user account or a group by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to bind an auditing policy to a user account or a group:

- bind audit <logtype> user <userName> **-policy** <policyname> [**-priority** <priority]
- bind audit <logtype> group <groupName> **-policy** <policyname> [**-priority** <priority]

Example

```
> bind audit nslogPolicy user aaa-user-1 -policyName Audit-Pol-1 -priority 1000
Done
```

To unbind a globally bound auditing policy by using the NetScaler command line

At the NetScaler command prompt, type the following commmands to unbind a globally-bound auditing policy:

unbind audit <logtype> global **-policy** <policyname>

Example

```
> unbind audit nslogPolicy global -policy Audit-Pol-1
Done
```

To unbind an auditing policy from an authentication virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind an auditing policy from an authentication virtual server:

unbind authentication vserver <authvsname> **-policy** <policyname>

Example

```
> unbind authentication vserver auth-vserver-1 -policyName Audit-Pol-1
Done
```

To unbind an auditing policy from a user account or a group by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to unbind an auditing policy from a user account or a group:

- unbind audit <logtype> user <userName> **-policy** <policyname>
- unbind audit <logtype> group <groupName> **-policy** <policyname>

Example

```
> unbind audit nslogPolicy group aaa-group-1 -policyName Audit-Pol-1
Done
```

To remove an auditing policy by using the NetScaler command line

First unbind the policy from all users and groups, and then, at the NetScaler command prompt, type the following command to remove an auditing policy:

rm audit <logtype> <policyname>

Parameters for configuring auditing policies

logtype

Which type of log your policy uses. If your policy logs to nslog, type nslogPolicy. If your policy logs to an external syslog server, type syslogPolicy.

policyname

A name for the policy you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing policies.)

rule

A NetScaler classic expression that defines which requests to audit. If you do not specify a rule, audit policies default to ns_true, which logs all connections. For a complete description of NetScaler classic expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

action

The action associated with this policy. For auditing policies, the action is the name of the nslog or syslog server to which you want to direct connections that match the audit log policy. If you do not specify a server, the default nslog server or syslog server is used.

authvsname

The name of the authentication virtual server to which you are binding the policy.

username

The name of the user account to which you are binding the policy.

groupname

The name of the group to which you are binding the policy.

priority

The priority assigned to this auditing policy.

To configure and bind auditing policies by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, expand **Policies**, and then click **Auditing**.
2. In the details pane, do one of the following:
   - To create a new auditing policy, click **Add**.
   - To modify an existing auditing policy, select the policy, and then click **Open**.
3. In the **Create Auditing Policy** or **Configure Auditing Policy** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring auditing policies" as follows (asterisk indicates a required parameter):
   - Name*—policyname (Cannot be changed for a previously configured policy.)
   - Auditing type*—logtype (When creating auditing policies by using the configuration utility, you cannot specify a rule.)
   - Server*—action
4. Click **Create**, and then click **Close**. The policy that you created appears in the **Auditing Policies and Servers** page.
5. Click **OK**.
6. To globally bind an auditing policy, in the details pane, click **Global Bindings** and fill in the **Bind/Unbind Audit Policies to Global** dialog box.
   a. Select the name of the audit policy you want to globally bind.
   b. Click **OK**.

   A message appears in the status bar, stating that the policy has been configured successfully.
7. To bind an auditing policy to an authentication virtual server, in the navigation pane, click **Virtual Servers**, and add that policy to the authentication policies list.
   a. In the details pane, select the appropriate virtual server, and then click **Open**.
   b. Click the **Policies** tab.
   c. Click **Insert Policy**.
   d. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
   e. In the **Priority** column, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
   f. Click **OK**.
8. To bind an auditing policy to a user account or group, in the navigation pane, click **Users** or **Groups**, and add that policy to the user account list.

a. In the details pane, select the appropriate user account, and then click **Open**.

b. Click the **Policies** tab.

c. Click **Insert Policy**.

d. Choose the policy you want to bind to the group from the drop-down list.

e. In the **Priority** column, modify the default priority as needed to ensure that the policy is evaluated in the proper order.

f. Click **OK**.

A message appears in the status bar, stating that the policy has been configured successfully.

Session Settings

After you configure your authentication, authorization, and auditing profiles, you configure session settings to customize your user sessions. The session settings are:

The session timeout.

Controls the period after which the user is automatically disconnected and must authenticate again to access your intranet.

The default authorization setting.

Determines whether the NetScaler appliance will by default allow or deny access to content for which there is no specific authorization policy.

The single sign-on setting.

Determines whether the NetScaler appliance will log users onto all web applications automatically after they authenticate, or will pass users to the web application logon page to authenticate for each application.

The credential index setting.

Determines whether the NetScaler appliance will use primary or the secondary authentication credentials for single signon.

To configure the session settings, you can take one of two approaches. If you want different settings for different user accounts or groups, you create a profile for each user account or group for which you want to configure custom sessions settings. You also create policies to select the connections to which to apply particular profiles, and you bind the policies to users or groups. You can also bind a policy to the authentication virtual server that handles the traffic to which you want to apply the profile.

If you want the same settings for all sessions, or if you want to customize the default settings for sessions that do not have specific profiles and policies configured, you can simply configure the global session settings.

Session Profiles

To customize your user sessions, you first create a session profile. The session profile allows you to override global settings for any of the session parameters.

Note: The terms “session profile” and “session action” mean the same thing.

To create a session profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a session profile and verify the configuration:

- add tm sessionAction <actionname> **-sessTimeout** <timeout> **-defaultAuthorizationAction** <defaultaction> [**-SSO** (ON | OFF)] [**-ssoCredential** (PRIMARY | SECONDARY)]
- show tm sessionAction <name>

Example

```
> add tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
       Authorization action : ALLOW
       Session timeout: 30 minutes
 Done
```

To modify a session profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify a session profile and verify the configuration:

- set tm sessionAction <actionname> **-sessTimeout** <timeout> **-defaultAuthorizationAction** <defaultaction> [**-SSO** (ON | OFF)] [**-ssoCredential** (PRIMARY | SECONDARY)]
- show tm sessionAction

Example

```
> set tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
       Authorization action : ALLOW
       Session timeout: 30 minutes
 Done
```

To remove a session profile by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a session profile:

rm tm sessionAction <actionname>

Parameters for configuring session profiles

actionname

A name for the session action, or the name of the existing session action you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing session actions.)

sesstimeout

The session timeout in minutes. Possible values: 1-65536. Default: 30.

defaultAuthorizationAction

Whether to allow or deny access to resources by default when no specific policy overrides this setting. Possible values: ALLOW, DENY. Default: DENY

sso

Whether to use single sign-on or not for all web applications. Possible values: YES, NO. Default: NO.

ssocredential

Whether to use the primary or secondary authentication server for single sign-on credentials. Possible values: PRIMARY, SECONDARY. Default: PRIMARY.

To configure session profiles by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Session**.
2. In the details pane, click the **Profiles** tab.
3. On the **Profiles** tab, do one of the following:
   - To create a new session profile, click **Add**.
   - To modify an existing session profile, select the profile, and then click **Open**.
4. In the **Create TM Session Profile** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring session profiles" as follows (asterisk indicates a required parameter):
   - Name*—actionname (Cannot be changed for a previously configured session action.)
   - Session Time-out—sesstimeout
   - Default Authorization Action—defaultAuthorizationAction
   - Single Signon to Web Applications—sso
   - Credential Index—ssocredential
5. Click **Create** or **OK**, and then click **Close**. The session profile that you created appears in the **Session Policies and Profiles** pane.

Session Policies

After you create one or more session profiles, you create session policies and then bind the policies globally or to an authentication virtual server to put them into effect.

To create a session policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a session policy and verify the configuration:

- add tm sessionPolicy <policyname> <rule> <actionname>
- show tm sessionPolicy <name>

Example

```
> add tm sessionPolicy session-pol "URL == /*.gif" session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.gif'
       Action: session-profile
 Done
```

To modify a session policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify a session policy and verify the configuration:

- set tm sessionPolicy <policyname> [**-rule** <expression>] [**-action** <action>]
- show tm sessionPolicy <name>

Example

```
> set tm sessionPolicy session-pol "URL == /*.gif" session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.gif'
```

```
      Action: session-profile
Done
```

To globally bind a session policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind a session policy and verify the configuration:

bind tm global **-policyName** <policyname> [**-priority** <priority>]

Example

```
> bind tm global -policyName session-pol
 Done

> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.gif'
       Action: session-profile
       Policy is bound to following entities
       1) TM GLOBAL    PRIORITY : 0
 Done
```

To bind a session policy to an authentication virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following command to bind a session policy to an authentication virtual and verify the configuration:

bind authentication vserver <authvsname> **-policy** <policyname> [**-priority** <priority>]

Example

```
> bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -priority 1000
Done
```

To unbind a session policy from an authentication virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a session policy from an authentication virtual server and verify the configuration:

unbind authentication vserver <authvsname> **-policy** <policyname>

Example

```
> unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
Done
```

To unbind a globally bound session policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a globally-bound session policy:

unbind tm global **-policyName** <policyname>

Example

```
> unbind tm global -policyName Session-Pol-1
Done
```

To remove a session policy by using the NetScaler command line

First unbind the session policy from global, and then, at the NetScaler command prompt, type the following commands to remove a session policy and verify the configuration:

rm tm sessionPolicy <policyname>

Example

```
> rm tm sessionPolicy Session-Pol-1
Done
```

Parameters for configuring session policies

policyname

A name for the session policy, or the name of the existing session policy you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing session policies.)

rule

A NetScaler classic expression that defines the requests to select. If you do not specify a rule, session profiles default to ns_true, which selects all connections. For a complete description of NetScaler classic expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

actionname

The name of the session profile to apply to connections that match this policy.

priority

The priority assigned to this session policy.

authvsname

The name of the authentication virtual server to which you are binding this session policy.

To configure and bind session policies by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Session**.
2. In the details pane, on the **Policies** tab, do one of the following:
   - To create a new session policy, click **Add**.
   - To modify an existing session policy, select the policy, and then click **Open**.
3. In the Create Session Policy or Configure Session Policy dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring session policies" as follows (asterisk indicates a required parameter):
   - Name*—policyname (Cannot be changed for a previously configured session policy.)
   - Request Profile*—actionname
   - Expression*—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
4. Click **Create** or **OK**, and then click **Close**. The policy that you created appears in the details pane of the **Session Policies and Profiles** page.
5. To globally bind an auditing policy, in the details pane, click **Global Bindings** and fill in the **Bind/Unbind Session Policies to Global** dialog box.
   a. Select the name of the session policy you want to globally bind.
   b. Click **OK**.
6. To bind a session policy to an authentication virtual server, in the navigation pane, click Virtual Servers, and add that policy to the policies list.
   a. In the details pane, select the appropriate virtual server, and then click **Open**.
   b. Click the **Policies** tab.
   c. Click **Insert Policy**.
   d. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
   e. In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
   f. Click **OK**.

A message appears in the status bar, stating that the policy has been configured successfully.

Global Session Settings

In addition to or instead of creating session profiles and policies, you can configure global session settings. These settings control the session configuration when there is no explicit policy overriding them.

To configure the session settings by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure the global session settings and verify the configuration:

set tm sessionParameter [**-sessTimeout** <timeout>] [**-defaultAuthorizationAction** <defaultaction>] [**-SSO** <sso>] [**-ssoCredential** <ssocredentials>]

Example

```
> set tm sessionParameter -sessTimeout 30
  Done
> set tm sessionParameter -defaultAuthorizationAction DENY
  Done
> set tm sessionParameter -SSO ON
  Done
> set tm sessionParameter -ssoCredential PRIMARY
  Done
```

Parameters for configuring global session settings

sessTimeout

The timeout for user sessions, as an integer value representing a number of minutes.

defaultAuthorizationAction

The default authorization action for a user request, when no specific policy is available. Possible values: ALLOW, DENY. Default: DENY.

sso

Whether to allow a user to authenticate once and then have access to all web applications on your intranet, or to require authentication for each application. Possible values: ON, OFF. Default: OFF.

ssoCredential

If single signon is enabled, which group of credentials to use for authentication. Possible values: Primary, Secondary. Default: Primary.

To configure the session settings by using the configuration utility

1. In the navigation pane, click **AAA - Application Traffic** to display the top-level details pane.
2. In the details pane, under **Settings**, click **Change global settings**.
3. In the **Global Session Settings** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring global session settings" as follows
   - Session Time-out—sessTimeout
   - Default Authorization Action—defaultAuthorizationAction
   - Single Sign-on to Web Applications—sso
   - Credential Index—ssoCredential
4. Click **OK**, and then click **Close**. A message appears in the status bar, stating that the settings have been configured successfully.

Traffic Settings

To use forms-based single sign-on (SSO) for your protected applications, you configure that feature in the Traffic settings. SSO enables your users to log on once to access all protected applications on your LAN, rather than requiring them to log on separately to access each one. Forms-based SSO allows you to use a web form of your own design as the sign-on method instead of a generic pop-up window. You can therefore put you company logo and other information you might want your users to see on the logon form.

To configure forms-based SSO, you first create a form SSO profile. Next, you create a traffic profile and link it to the form SSO profile you created. Next, you create a policy, link it to the traffic profile. Finally, you bind the policy globally or to an authentication virtual server to put your configuration into effect.

Form SSO Profiles

To enable and configure forms-based SSO, you first create an SSO profile.

Note: In this feature, the terms “profile” and “action” mean the same thing.

To create a form SSO profile by using the NetScaler command line

At the NetScaler command prompt, type:

add tm formSSOAction <name> **-actionURL** <URL> **-userField** <string> **-passwdField** <string> **-ssoSuccessRule** <expression> [**-nameValuePair** <string>] [**-responsesize** <positive_integer>] [**-nvtype** (STATIC | DYNAMIC)] [**-submitMethod** (GET | POST)

Example

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
-sessTimeout 10 -defaultAuthorizationAction ALLOW
```

To modify a form SSO by using the NetScaler command line

At the NetScaler command prompt, type:

set tm formSSOAction <name> **-actionURL** <URL> **-userField** <string> **-passwdField** <string> **-ssoSuccessRule** <expression> [**-nameValuePair** <string>] [**-responsesize** <positive_integer>] [**-nvtype** (STATIC | DYNAMIC)] [**-submitMethod** (GET | POST)

Example

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
```

-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
–sessTimeout 10 -defaultAuthorizationAction ALLOW

To remove a form SSO profile by using the NetScaler command line

At the NetScaler command prompt, type:

rm tm formSSOAction <name>

Example

rm tm sessionAction SSO-Prof-1

Parameters for configuring form SSO profiles

name

A name for the SSO action, or the name of the existing SSO action you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing SSO actions.)

actionURL

The URL where the SSO logon form is located.

userField

The form field where the user types in the user ID or login.

passwdField

The form field where the user types in the password.

ssoSuccessRule

An expression that describes the action that this profile should take when invoked by a policy.

responsesize

The number of bytes to allow for the complete response size. Responses that exceed this value are blocked.

nameValuePair

The userField value, followed by a space, followed by the passwdField value. When typing this at the NetScaler command line, you should enclose the nameValuePair in straight double quotes.

nvtype

Whether the name/value pair is static or dynamic. Possible values: STATIC, DYNAMIC. Default: STATIC.

submitMethod

HTTP method used by the SSO logon form to send the logon credentials to the logon server. Possible values: GET, POST. Default: POST.

To configure form SSO profiles by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Traffic**.
2. In the details pane, click the **Form SSO Profiles** tab.
3. On the **Form SSO Profiles** tab, do one of the following:
   - To create a new form SSO profile, click **Add**.
   - To modify an existing form SSO profile, select the profile, and then click **Open**.
4. In the **Create Form SSO Profile** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in “Parameters for configuring form SSO profiles” as follows (an asterisk indicates a required parameter):
   - **Name***—name (Cannot be changed for a previously configured session action.)
   - **Action URL***—actionURL
   - **User Name Field***—userField
   - **Password Field***—passField
   - **SSO Success Rule***—ssoSuccessRule
   - **Name Value Pair**—nameValuePair
   - **Response Size**—responsesize
   - **Extraction**—nvtype
   - **Submit Method**—submitMethod
5. Click **Create** or **OK**, and then click **Close**. The form SSO profile that you created appears in the **Traffic Policies**, **Profiles**, and **Form SSO Profiles** pane.

Traffic Profiles

After creating at least one form sso profile, you must next create a traffic profile.

Note: In this feature, the terms “profile” and “action” mean the same thing.

To create a traffic profile by using the NetScaler command line

At the NetScaler command prompt, type:

add tm trafficAction <name> [**-appTimeout** <mins>] [**-SSO** (ON | OFF) [**-formSSOAction** <string>]]

Example

add tm trafficAction Traffic-Prof-1 -appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1

To modify a session profile by using the NetScaler command line

At the NetScaler command prompt, type:

set tm trafficAction <name> [**-appTimeout** <mins>] [**-SSO** (ON | OFF) [**-formSSOAction** <string>]]

Example

set tm trafficAction Traffic-Prof-1 -appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1

To remove a session profile by using the NetScaler command line

At the NetScaler command prompt, type:

rm tm trafficAction <actionname>

Example

rm tm trafficAction Traffic-Prof-1

Parameters for configuring form traffic profiles

name

A name for the traffic action, or the name of the existing traffic action you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing SSO actions.)

appTimeout

The application timeout in minutes. Possible values: 1-65536. Default: 30.

SSO

Whether to enable or disable SSO. Possible values: ON, OFF. Default: OFF.

formSSOAction

The name of the form SSO profile to use.

To configure traffic profiles by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Traffic**.
2. In the details pane, click the **Profiles** tab.
3. On the **Profiles** tab, do one of the following:
   - To create a new traffic profile, click **Add**.
   - To modify an existing traffic profile, select the profile, and then click **Open**.
4. In the **Create Traffic Profile** or **Configure Traffic Profile** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in “Parameters for configuring form traffic profiles” as follows (an asterisk indicates a required parameter):
   - **Name***—name (Cannot be changed for a previously configured session action.)
   - **AppTimeout**—appTimeout
   - **Single Sign-On**—SSO
   - **Form SSO Action**—formSSOAction
5. Click **Create** or **OK**, and then click **Close**. The traffic profile that you created appears in the **Traffic Policies**, **Profiles**, and **Form SSO Profiles** pane.

Traffic Policies

After you create one or more form SSO and traffic profiles, you create traffic policies and then bind the policies, either globally or to an authentication virtual server, to put them into effect.

To create a traffic policy by using the NetScaler command line

At the NetScaler command prompt, type:

add tm trafficPolicy <policyName> "<rule>" <action>

Example

add tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true")" Traffic-Prof-1

To modify a traffic policy by using the NetScaler command line

At the NetScaler command prompt, type:

set tm trafficPolicy <policyName> "<rule>" <action>

Example

set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true")" Traffic-Prof-1

To globally bind a traffic policy by using the NetScaler command line

At the NetScaler command prompt, type:

bind tm global **-policyName** <policyName> [**-priority** <priority>]

Example

bind tm global -policyName Traffic-Pol-1

To bind a traffic policy to an authentication virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind authentication vserver <authvsname> **-policy** <policyName> [**-priority** <priority>]

Example

bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -priority 1000

To unbind a globally bound traffic policy by using the NetScaler command line

At the NetScaler command prompt, type:

unbind tm global **-policyName** <policyname>

Example

unbind tm global -policyName Traffic-Pol-1

To unbind a traffic policy from an authentication virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

unbind authentication vserver <authvsname> **-policy** <policyname>

Example

unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1

To remove a traffic policy by using the NetScaler command line

First unbind the session policy from global, and then, at the NetScaler command prompt, type:

rm tm trafficPolicy <policyname>

Example

rm tm trafficPolicy Traffic-Pol-1

Parameters for configuring form traffic profiles

policyName

A name for the traffic policy, or the name of the existing traffic policy you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing SSO actions.)

rule

A NetScaler advanced expression that defines the requests to select. For a complete description of NetScaler advanced expressions, see the *Citrix NetScaler Policy Configuration & Reference Guide* at http://support.citrix.com/article/CTX128673

actionname

The name of the traffic profile to apply to connections that match this policy.

priority

The priority assigned to this traffic policy.

authvsname

The name of the authentication virtual server to which you are binding this traffic policy.

To configure and bind traffic policies by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, then expand **Policies**, and then click **Traffic**.
2. In the details pane, do one of the following:
   - To create a new session policy, click **Add**.
   - To modify an existing session policy, select the policy, and then click **Open**.
3. In the Create Traffic Policy or Configure Traffic Policy dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring form traffic profiles" as follows (an asterisk indicates a required parameter):
   - **Name***—policyName (Cannot be changed for a previously configured session policy.)
   - **Profile***—actionName
   - **Expression**—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
4. Click **Create** or **OK**, and then click **Close**. The policy that you created appears in the details pane of the **Session Policies and Profiles** page.
5. To globally bind a traffic policy, in the details pane, click **Global Bindings** and fill in the **Bind/Unbind Session Policies to Global** dialog box.
   a. Select the name of the traffic policy you want to globally bind.
   b. Click **OK**.
6. To bind a traffic policy to an authentication virtual server, in the navigation pane, click **Virtual Servers**, and add that policy to the policies list.
   a. In the details pane, select the appropriate virtual server, and then click **Open**.
   b. Click the **Policies** tab
   c. Click Insert **Policy**.
   d. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
   e. In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
   f. Click **OK**.

Authenticating with Client Certificates

Web sites that contain sensitive content, such as online banking web sites or Web sites with employee personal information, sometimes require client certificates for authentication. To configure AAA to authenticate users on the basis of client-side certificate attributes, you first enable client authentication on the traffic management virtual server and bind the root certificate to the authentication virtual server. Then, you implement one of two options. You can configure the default authentication type on the authentication virtual server as CERT, or you can create a certificate action that defines what the NetScaler appliance must do to authenticate users on the basis of a client certificate. In either case, your authentication server must support CRLs. You configure the NetScaler appliance to extract the user name from the SubjectCN field or another specified field in the client certificate.

When the user tries to log in to an authentication virtual server for which an authentication policy is not configured, and a global cascade is not configured, the user name information is extracted from the specified field of the certificate. If the required field is extracted, the authentication succeeds. If the user does not provide a valid certificate during the SSL handshake, or if the user name extraction fails, authentication fails. After it validates the client certificate, the NetScaler appliance presents a login page to the user.

The following procedures assume that you have already created a functioning AAA configuration, and therefore they explain only how to enable authentication by using client certificates. These procedures also assume that you have obtained your root certificate and client certificates and have placed them on the NetScaler appliance in the /nsconfig/ssl directory.

To configure the AAA client certificate parameters by using the NetScaler command line

At the NetScaler command prompt, type the following commands in the order shown and verify the configuration:

- add ssl certKey <certkeyName> **-cert** <certFile> **-key** <keyFile> **-password -inform** <inform> **-expiryMonitor** <expiryMonitor> **-notificationPeriod** <notificationPeriod>
- bind ssl certKey <vservername>@ **-vServer** <certkeyName> **-CA -crlCheck** Mandatory
- set aaa parameter **-defaultAuthType** CERT
- set aaa certParams **-userNameField** "Subject:CN"

Parameters for authentication with client certificates

certkeyName

The name of the certificate-key pair.

certFile

The name of the file containing the certificate.

keyFile

The name of the file containing the private key.

password

The password to the private key. If you are entering this at the NetScaler command line, you simply include the -password parameter, and then, after entering the command, enter the actual password when prompted. If you are using the configuration utility, you type the password in the appropriate text box.

inform

The format of the certificate-key pair. Possible values: DER, PEM. Default: PEM.

expiryMonitor

Whether to monitor the certificate-key pair for expiration. Possible values: ENABLED, DISABLED. Default: ENABLED.

notificationPeriod

The period, in days before the certificate-key pair expires, during which the NetScaler appliance should notify the administrator of the impending expiration.

vservername

The name of the authentication virtual server to which to bind the root certificate.

To configure the AAA client certificate parameters by using the configuration utility

1. In the navigation pane, expand **AAA - Application Traffic**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure to handle client certificate authentication, and then click **Open**.
3. In the **Configure Virtual Server (Authentication)** dialog box, in the **Certificates** tab, click **Install**.
4. In the **Install Certificate** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for authentication with client certificates" as follows (asterisk indicates a required parameter):
   - Certificate-Key Pair Name*—certkeyName
   - Certificate File Name—certFile
   - Private Key File Name—keyFile
   - Password—password
   - Certificate Format—inform
   - Notify When Expires—expiryMonitor
   - Notification Period—notificationPeriod
5. Click **Install**, and then click **Close**.
6. In the **Configure Virtual Server (Authentication)** dialog box, in the **Available** list, select the root certificate.
7. Click **Add as CA**.
8. Click **OK**.
9. In the navigation pane, expand **Policies**, and then click **Authentication**.
10. In the details pane, select the policy you want to configure to handle client certificate authentication, and then click **Open**.
11. In the **Authentication Type** drop-down list, select **CERT**.
12. In the **Server** drop-down list, select the virtual server you just configured to handle client certificate authentication.
13. Click **OK**. A message appears in the status bar, stating that the configuration completed successfully.

Configuring AAA with Commonly Used Protocols

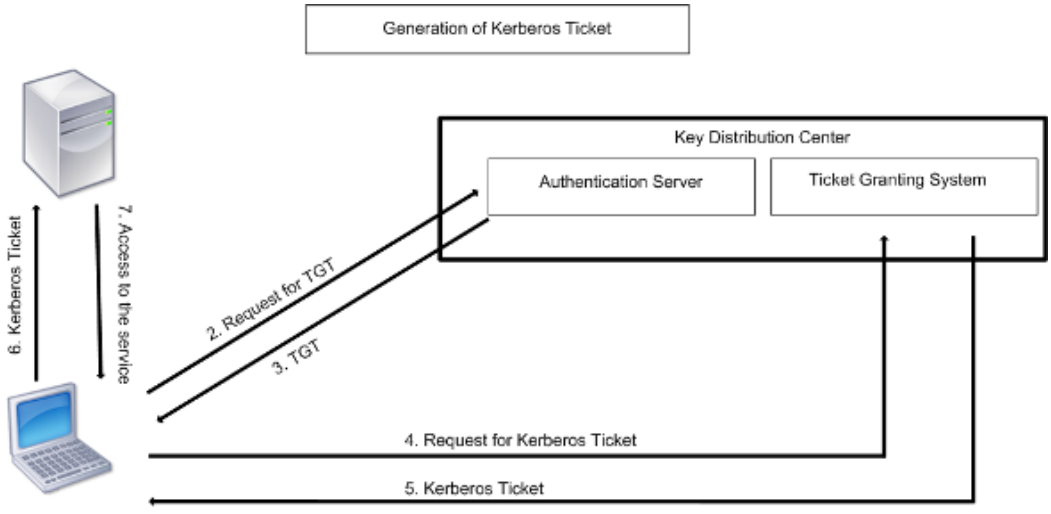
Configuring the NetScaler for Authentication, Authorization, and Auditing (AAA) needs a specific setup on the NetScaler and client's browsers. The configuration varies with the protocol used for AAA.

For more information about configuring the NetScaler for Kerberos authentication, see Handling Authentication, Authorization and Auditing with Kerberos/NTLM.

Handling Authentication, Authorization and Auditing with Kerberos/NTLM

Kerberos, a computer network authentication protocol, provides secure communication over the Internet. Designed primarily for client-server applications, it provides for mutual authentication by which the client and server can each ensure the other's authenticity. Kerberos uses a trusted third party, referred to as Key Distribution Center (KDC). A KDC consists of an Authentication Server (AS), which authenticates a user, and a Ticket Granting Server (TGS).

Each entity on the network (client or server) has a secret key that is known only to itself and the KDC. The knowledge of this key implies authenticity of the entity. For communication between two entities on the network, the KDC generates a session key, referred to as the *Kerberos ticket* or *service ticket*. The client makes a request to the AS for credentials for a specific server. The client then receives a ticket, referred to as Ticket Granting Ticket (TGT). The client then contacts the TGS, using the TGT it received from the AS to prove its identity, and asks for a service. If the client is eligible for the service, the TGS issues a Kerberos ticket to the client. The client then contacts the server hosting the service (referred to as the service server), using the Kerberos ticket to prove that it is authorized to receive the service. The Kerberos ticket has a configurable lifetime. The client authenticates itself with the AS only once. If it contacts the physical server multiple times, it reuses the AS ticket.



The following figure shows the basic functioning of the Kerberos protocol.Figure 1. **Functioning of Kerberos**

See

- Implementation of Kerberos authentication on the NetScaler
- Configuration of Kerberos authentication on the NetScaler

- Configuration of Kerberos from the command line
- Configuration of Kerberos from the configuration utility
- Configuration of Kerberos authentication on the client

Kerberos authentication has the following advantages:

- Faster authentication. When a physical server gets a Kerberos ticket from a client, the server has enough information to authenticate the client directly. It does not have to contact a domain controller for client authentication, and therefore the authentication process is faster.
- Mutual authentication. When the KDC issues a Kerberos ticket to a client and the client uses the ticket to access a service, only authenticated servers can decrypt the Kerberos ticket. If the virtual server on the NetScaler is able to decrypt the Kerberos ticket, you can conclude that both the virtual server and client are authenticated. Thus, the authentication of the server happens along with the authentication of the client.
- Single sign-on between Windows and other operating systems that support Kerberos.

Kerberos authentication may have the following disadvantages:

- Kerberos has strict time requirements; the clocks of the involved hosts must be synchronized with the Kerberos server clock to ensure that the authentication does not fail. You can mitigate this disadvantage by using the Network Time Protocol daemons to keep the host clocks synchronized. Kerberos tickets have an availability period, which you can configure.
- Kerberos needs the central server to be available continuously. When the Kerberos server is down, no one can log in. You can mitigate this risk by using multiple Kerberos servers and fallback authentication mechanisms.
- Because all the authentication is controlled by a centralized KDC, any compromise in this infrastructure, such as the user's password for a local workstation being stolen, can allow an attacker to impersonate any user. You can mitigate this risk to some extent by using only a desktop machine or laptop that you trust, or by enforcing preauthentication by using a hardware-token.

How NetScaler Implements Kerberos Authentication

Note: Kerberos/NTLM authentication is supported only in the NetScaler 9.3 nCore release or later, and it can be used only for AAA traffic management (AAA-TM) virtual servers.

NetScaler handles the components involved in Kerberos authentication in the following way:

Key Distribution Center (KDC)

In the Windows 2000 Server or later versions, the Domain Controller and KDC are part of the Windows Server. If the Windows Server is UP and running, it indicates that the Domain Controller and KDC are configured. The KDC is also the Active Directory server.

Note: All Kerberos interactions are validated with the Windows Kerberos Domain Controller.

Authentication Service and Protocol Negotiation

NetScaler supports Kerberos authentication on the AAA-TM authentication virtual servers. If the Kerberos authentication fails, the NetScaler uses the NTLM authentication.

By default, Windows 2000 Server and later Windows Server versions use Kerberos for AAA. If you create an authentication policy with NEGOTIATE as the authentication type, the NetScaler tries to use the Kerberos protocol for AAA and if the client's browser fails to receive a Kerberos ticket, the NetScaler uses the NTLM authentication. This process is referred to as *negotiation*.

The client may fail to receive a Kerberos ticket in any of the following cases:

- Kerberos is not supported on the client.
- Kerberos is not enabled on the client.
- The client is in a domain other than that of the KDC.
- The Access Directory on the KDC is not accessible to the client.

For Kerberos/NTLM authentication, the NetScaler does not use the data that is present locally on the NetScaler appliance.

Authorization

The traffic management virtual server can be a load balancing virtual server or a content switching virtual server.

Auditing

The NetScaler appliance supports auditing of Kerberos authentication with the following audit logging:

- Complete audit trail of the traffic management end-user activity
- SYSLOG and high performance TCP logging
- Complete audit trail of system administrators
- All system events
- Scriptable log format

Supported Environment

Kerberos authentication does not need any specific environment on the NetScaler. The client (browser) should have the support for Kerberos authentication.

High Availability

In a high availability setup, only the active NetScaler joins the domain. In case of a failover, the NetScaler lwagent daemon joins the secondary NetScaler appliance to the domain. No specific configuration is required for this functionality.

Kerberos Authentication Process

The following figure shows a typical process followed for Kerberos authentication in the NetScaler environment.

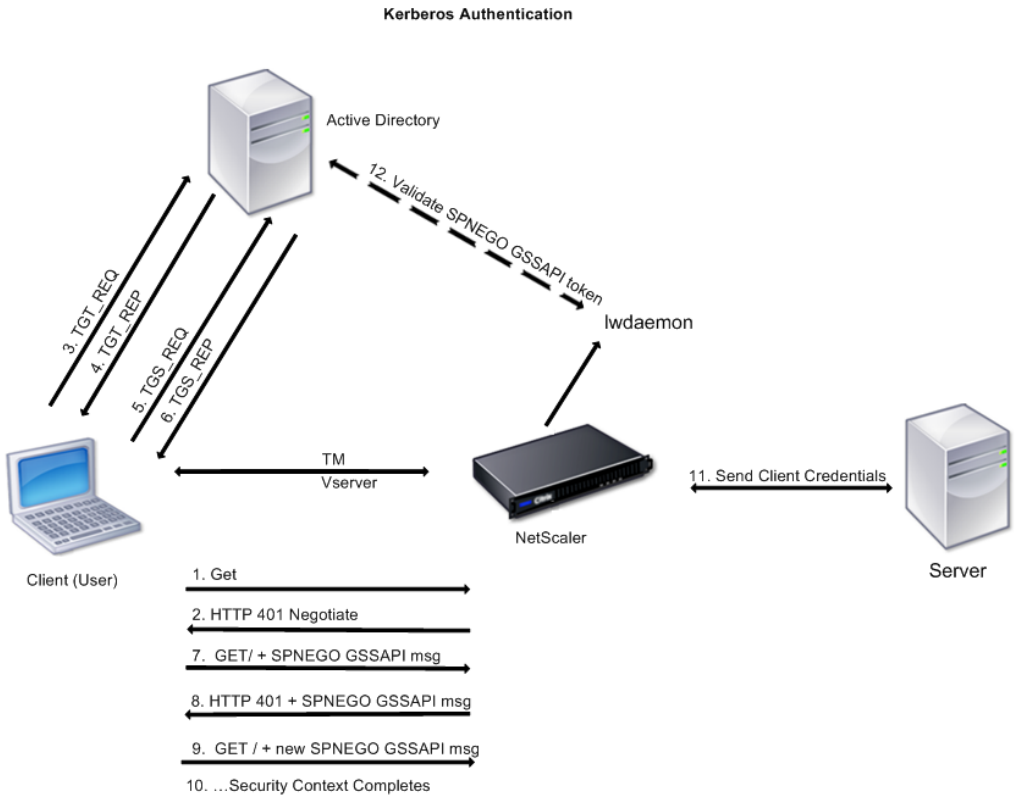


Figure 2. Kerberos Authentication Process on NetScaler

The Kerberos authentication occurs in the following stages:

Client authenticates itself to the KDC.

1. The NetScaler appliance receives a request from a client.
2. The traffic management (load balancing or content switching) virtual server on the NetScaler sends a challenge to the client.
3. To respond to the challenge, the client gets a Kerberos ticket.
    - The client requests the Authentication Server of the KDC for a ticket-granting ticket (TGT) and receives it. (See 3, 4 in the figure, Kerberos Authentication Process.)
    - The client sends the TGT to the Ticket Granting Server of the KDC and receives a Kerberos ticket. (See 5, 6 in the figure, Kerberos Authentication Process.)

Note: The above authentication process is not necessary if the client already has a Kerberos ticket whose lifetime has not expired. In addition, clients such as Web Services, .NET, or J2EE which support SPNEGO get a Kerberos ticket for the target server, create

an SPNEGO token, and insert it in the HTTP header when they send an HTTP request. They do not go through the client authentication process.

Client requests a service.

1. The client sends Kerberos ticket containing the SPNEGO token and the HTTP request to the traffic management virtual server on the NetScaler. The SPNEGO token has the necessary GSSAPI data.

2. The NetScaler establishes a security context between the client and the NetScaler. If the NetScaler cannot accept the data provided in the Kerberos ticket, the client is asked to get a different ticket. This cycle repeats till the GSSAPI data is acceptable and the security context is established. The traffic management virtual server on the NetScaler acts as an HTTP proxy between the client and the physical server.

NetScaler completes the authentication.

1. After the security context is complete, the traffic management virtual server validates the SPNEGO token.

2. From the valid SPNEGO token, the virtual server extracts the user ID and GSS credentials, and passes them to the authentication daemon.

3. A successful authentication completes the Kerberos authentication.

Configuring Kerberos Authentication on the NetScaler

To configure Kerberos authentication, the NetScaler and the client browser should have some specific configuration.

Configuration on the NetScaler

1. Enable the **Authentication, Authorization, and Auditing (AAA)** feature on the NetScaler appliance.

2. On the Active Directory, add a user for Kerberos authentication, map the HTTP service to this user, and generate a keytab file and import it to the NetScaler appliance. You can map more than one service if the Kerberos authentication is required for more than one service. The keytab file should contain entries for every service that is bound to the traffic management virtual server on the NetScaler. The keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. The authentication details of all the services are stored in a single keytab file on the NetScaler.

3. Add a DNS server.

   Note: It is necessary that the NetScaler obtains the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, it is recommended to configure the NetScaler with a DNS server. A less preferred alternative is to create a static DNS entry.

4. Create an authentication negotiation policy with a negotiation action.
5. Configure an authentication server and bind the authentication policy to the authentication virtual server.
6. Configure an authentication service and a traffic management virtual server, and bind the service to the virtual server. You can use a load balancing or content switching virtual server.
7. Verify the configuration.

Configuration of Kerberos from the command line

The following sections give instructions to configure Kerberos from the command line.

To enable Authentication, Authorization, and Auditing (AAA) by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable AAA and verify the configuration:

- enable feature **AAA**
- show ns feature

Example

```
> enable feature aaa
 Done
> show ns feature
 Feature Acronym Status
 ------- ------- ------
1) Web Logging WL ON
 …
3) Load Balancing LB ON
4) Content Switching CS ON
5) Cache Redirection CR ON
 …
14) SSL VPN SSLVPN ON
15) AAA AAA ON
 …
26) CloudBridge CloudBridge OFF
 Done
```

To generate a keytab file and import it to the NetScaler by using the NetScaler command line

To generate the keytab file and import it onto the NetScaler, follow the procedure given below:

1. In the Active Directory, create a user for Kerberos authentication. For example,

   net user Kerb-SVC-Account **freebsd!@#456 /add**

   In the User Properties section, ensure the following settings:

   a. The **Change password at next logon option** is not selected.

   b. The **Password does not expire** option is selected.

   Map the HTTP service to the above user and export the keytab file.

   For example, run the following command from the NetScaler command line interface:

   ktpass **/out keytabfile /princ** HTTP/owa.newacp.com@NEWACP.COM **/pass** freebsd!@#456 **/mapuser** newacp\dummy **/ptype** KRB5_NT_PRINCIPAL

   If you want to map more services, repeat the above command for every service. You can give the same name or different names for the output file.

3. Get the keytab file on the NetScaler through FTP or any other means.

   Run the ktutil utility on the NetScaler and verify the keytab file. The keytab file has the entry for the HTTP service after it is imported.

   Example

   ```
   root@ns# ktutil
   ktutil: rkt /var/keytabfile
   ktutil: list
   slot KVNO Principal
   ----------------------------------------------------------------------

   ktutil: wkt /etc/ krb5.keytab
   ktutil: list
   slot KVNO Principal
   ---- ---- ---------------------------------------------------------------
   1 2 HTTP/owa.newacp.com@NEWACP.COM
   ktutil: quit
   ```

To add a DNS server by using the NetScaler command line

At the NetScaler command prompt, type the following command:

add dns nameserver <dnsIpAddress>

Note: Alternatively, you can add static host entries or use any other means so that the NetScaler can resolve the FQDN name of the domain controller to an IP address.

Example

add dns nameserver 1.2.3.4

To create a negotiation policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add authentication negotiateAction <negotiateActionName> **-domain** <domainName> **-domainUser** <domainUsername> **-domainUserPasswd** <domainUserPassword> **-encrypted**
- add authentication negotiatePolicy <negotiatePolicyName> <negotiatePolicyExpression> <negotiateActionName>

Example

add authentication negotiateAction negact -domain newacp.com -domainUser Administrator -domainUserPas
add authentication negotiatePolicy negopol ns_true negact

To create an authentication virtual server and bind the negotiation policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add authentication vserver <authVserverName> **SSL** <ipAuthVserver> **443** **-authenticationDomain** <domainName>
- bind authentication vserver <authVserverName> **-policy** <negotiatePolicyName>

Example

add authentication vserver authen1 SSL 10.102.113.166 443 -authenticationDomain newacp.com
add ssl certKey cert1 -cert "/nsconfig/ssl/complete/server/server_rsa_2048.pem" -key "/nsconfig/ssl/compl
bind ssl vserver authen1 -certkeyName cert1
bind authentication vserver authen1 -policy negopol

To create a traffic management virtual server and service, and bind the service by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add service <serviceName> <ipBackendWebserver> HTTP 80
- add lb vserver <lbVserverName> **SSL** <ipAddressLbVserver> **443 -authn401** ON **-authnVsName** <authVserverName>
- bind lb vserver <lbVserverName> <serviceName>

Note: Use a similar procedure for using a content switching virtual server as the traffic management virtual server.

Example

```
add service svc1 10.217.28.92 HTTP 80
 add lb vserver v2 HTTP 10.102.113.164 80 -persistenceType NONE -cltTimeout 180 -authn401 ON -authnVsNa
bind lb vserver v2 svc1
```

To verify the configuration from command line

1. Access the load balancing virtual server, using the FQDN. For example, http://owa.newacp.com.
2. View the AAA session on the NetScaler. show aaa session

Example

```
ClientIp (ClientPort) ->ServerIp(ServerPort)
 ---------------------------------------------
 PE id : 4
 User name: john.smith@NEWACP.COM Session Type: TM
Done
```

Parameters for configuring Kerberos authentication

For configuring negotiation policy

For configuring the DNS server

dnsIpAddress

The IP address of the name server that is used to resolve domain names to IP addresses.

negotiateActionName

The name of the negotiate action associated with the negotiate policy.

domainName

The fully qualified domain name in which the client and KDC are present.

domainUsername

The user name of the user who can access the domain.

domainUserPassword

The password of the user who can access the domain.

negotiatePolicyName

A name for the policy you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing policies.)

negotiatePolicyExpression

A policy expression that defines the requests to be authenticated.

For configuring authentication virtual server

authVserverName

A name for the new authentication virtual server. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed after the virtual server is created.)

ipAuthVserver

The IP address of the authentication virtual server.

domainName

The fully qualified domain name in which the client and KDC are present. This domain is assigned to the authentication virtual server.

negotiatePolicyName

A name for the policy you are creating. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. (Cannot be changed for existing policies.)

For configuring traffic management virtual server and service

serviceName

Name of the service. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ipBackendWebServer

IP address of the Web server used in backend communication.

lbVserverName

Name of the load balancing virtual server used as the traffic management virtual server.

csVserverName

Name of the content switching virtual server used as the traffic management virtual server.

ipAddressLbVserver

IP address of the load balancing virtual server.

ipAddressCsVserver

IP address of the content switching virtual server.

authVserverName

Name of the authentication virtual server associated with the traffic management virtual server.

Configuration of Kerberos from the configuration utility

The following sections give instructions to configure Kerberos from the configuration utility.

To enable Authentication, Authorization, and Auditing (AAA) on NetScaler by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Configure basic features**.
3. In the **Configure Basic Features** dialog box, select the **Authentication, Authorization and Auditing** check box.
4. Click **OK**.
5. In the confirmation dialog box, click **Yes**. A message appears in the status bar to indicate that the feature is enabled.

To generate a keytab file and import it to the NetScaler

You can generate the keytab file and import it onto the NetScaler only from the command line. See To generate a keytab file and import it to the NetScaler by using the command line.

To add a DNS server by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Name Servers**.
2. In the details pane, click **Add**.
3. In the **IP Address** box, type the IP address.
4. Click **Create**, and then **Close**.
5. Verify that the details pane shows the newly added DNS server.

To create a negotiation policy by using the NetScaler configuration utility

1. In the navigation pane, expand **AAA-Application Traffic**, expand **Policies**, and then click **Authentication**.
2. In the details pane, on the **Policies** tab, click **Add**.
3. In the **Create Authentication Policy** dialog box, set the following parameters:
   - **Name**
   - **Authentication Type** - Select **NEGOTIATE**.
   - **Server** - Select an existing server from the dropdown list. To add a new authentication server, click **New…**, and in the **Create Authentication Server** dialog box, set the following parameters:
     - **Domain Name**
     - **User Name**
     - **Password**
     - **Confirm Password** - Retype the password.
   - **Expression** - In the **Named Expression** list, select **General** and select **True Value** from the dropdown list, and then click **Add Expression**.
4. Click **Create**, and then click **Close**.
5. Verify that the policy you created appears in the **Authentication Policies and Servers** pane.

To create an authentication virtual server and bind the negotiation policy by using the NetScaler configuration utility

1. In the navigation pane, expand **AAA - Application Traffic** and click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Authentication)** dialog box, set the following parameters:
   - **Name**
   - **IP Address**
   - **Protocol** - Select **SSL**
   - **Domain** - Type the fully qualified domain name added while creating the keytab file.

   Note: For AAA, the protocol must be SSL protocol and port must be 443. Therefore, these options are not provided.
4. On the **Authentication** tab, click **Insert Policy**. In the **Authentication Policies** group, from the **Policy Name** dropdown list, select the negotiate authentication policy you added for Kerberos authentication.
5. On the **Certificates** tab, select an SSL certificate from the list of available certificates, and then click **Add**. If the certificate you want to bind is not displayed in the **Available Certificates** list, click **Install...**, and then select the certificate file.
6. Click **Create**, and then click **Close**. The new authentication virtual server appears in the **Authentication Virtual Servers** pane.

To create a traffic management virtual server and service, and bind the service by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, set the following parameters:
   - **Service Name**
   - **Server**
   - **Protocol** - Select **HTTP**.
   - **Port** - Select **80**.
4. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
5. In the details pane, click **Add**.
6. In the **Create Virtual Server (Load balancing)** dialog box, set values for the following parameters:
   - **Name**
   - **IP Address**
   - **Protocol**
   - **Port**
7. In the **Create Virtual Server (Load balancing)** dialog box, on the **Services** tab, select the service you created in Step 3 to Step 5.
8. In the **Create Virtual Server (Load balancing)** dialog box, on the **Advanced** tab, expand **Authentication Settings**, and then select the **401 Based Authentication** check box.
9. Click **Create**, and then click **Close**. The new load balancing virtual server appears in the **Load Balancing Virtual Servers** pane.
10. In the details pane, verify the settings of the virtual server.

    Note: Use a similar procedure to create a content switching virtual server.

    Note: For more information, see *Citrix NetScaler Traffic Management Guide* available at http://support.citrix.com/article/CTX128670.see *Setting up basic load balancing.*

To verify the configuration

1. Access the load balancing virtual server, using the FQDN. For example, http://owa.newacp.com.

2. View the AAA session on the NetScaler. show aaa session

Example

```
ClientIp (ClientPort) ->ServerIp(ServerPort)
 ---------------------------------------------
 PE id : 4
 User name: john.smith@NEWACP.COM Session Type: TM
Done
```

Configuring Kerberos Authentication on the Client

Kerberos support must be configured on the browser to use Kerberos for authentication. You can use any Kerberos-compliant browser. Instructions for configuring Kerberos support on Internet Explorer and Mozilla Firefox are given below. For other browsers, see the documentation of the browser.

To configure Internet Explorer for Kerberos authentication

1. In the **Tools** menu select **Internet Options**.

2. On the **Security** tab, click **Local Intranet**, and then click **Sites.**

3. In the **Local Intranet** dialog box, make sure that the **Automatically detect intranet network** option is checked and click **Advanced.**

4. In the **Local Intranet** dialog box, add the Web sites of the domains of the traffic management virtual server on the NetScaler. The specified sites will be considered as local intranet sites.

5. Click **Close** or **OK** to close the dialog boxes.

To configure Mozilla Firefox for Kerberos authentication

1. Make sure that you have Kerberos properly configured on your computer.
2. Type `about:config` in the URL bar.
3. In the filter text box, type `network.negotiate`.
4. Change network.negotiate-auth.delegation-uris to the domain that you want to add.
5. Change network.negotiate-auth.trusted-uris to the domain that you want to add.

   Note: If you are running Windows, you also need to enter `sspi` in the filter text box and change the network.auth.use-sspi option to `False`.

AppExpert

The following topics provide a conceptual reference and configuration instructions for the AppExpert feature, which simplifies configuration steps for the Citrix® NetScaler® appliance by using applications, application templates, Access Gateway applications, and entity templates.

AppExpert Applications and Templates	Simplify configuration steps for the Citrix® NetScaler® appliance by using applications, application templates, Access Gateway applications, and entity templates.
Access Gateway Applications	Describes how to configure Access Gateway policies for AppExpert applications, file shares, intranet subnets, and other network resources.
Entity Templates	Describes how to use entity templates to set up and configure individual NetScaler entities, such as a policy or virtual server. An entity template provides a specification and a set of defaults for the object.

AppExpert Applications and Templates

An AppExpert application is a collection of configuration information that you set up on the Citrix® NetScaler® appliance for securing and optimizing traffic for a Web application, such as Microsoft SharePoint. Managing AppExpert applications is simplified by a graphical user interface (GUI) that allows you to specify application traffic subsets and a distinct set of security and optimization policies for processing each traffic subset. Additionally, it consolidates all deployment tasks in one view, so you can quickly configure target IP addresses for clients and specify host servers.

Prebuilt application templates for widely used Web applications, such as Microsoft Outlook Web Access and Microsoft SharePoint, are available on the AppExpert Templates page of the Citrix Community Web site at http://community.citrix.com/display/ns/AppExpert+Templates.

Each prebuilt template provides you with an initial configuration for managing the associated Web application. You can customize prebuilt application templates for your organization. If a prebuilt application template does not suit your requirements, you can create a custom application without using a template.

Regardless of whether you use a prebuilt application template or you create a custom application, you can export the configuration to a template file. You can then share the template with other administrators or import the template to other NetScaler appliances that require a similar AppExpert application configuration.

To get started with an AppExpert application, you must first obtain the appropriate application template and import the template to the NetScaler appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components, statistics, and the Application Visualizer. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

AppExpert Applications and Templates

An AppExpert application is a collection of configuration information that you set up on the Citrix® NetScaler® appliance for securing and optimizing traffic for a Web application, such as Microsoft SharePoint. Managing AppExpert applications is simplified by a graphical user interface (GUI) that allows you to specify application traffic subsets and a distinct set of security and optimization policies for processing each traffic subset. Additionally, it consolidates all deployment tasks in one view, so you can quickly configure target IP addresses for clients and specify host servers.

Prebuilt application templates for widely used Web applications, such as Microsoft Outlook Web Access and Microsoft SharePoint, are available on the AppExpert Templates page of the Citrix Community Web site at http://community.citrix.com/display/ns/AppExpert+Templates.

Each prebuilt template provides you with an initial configuration for managing the associated Web application. You can customize prebuilt application templates for your organization. If a prebuilt application template does not suit your requirements, you can create a custom application without using a template.

Regardless of whether you use a prebuilt application template or you create a custom application, you can export the configuration to a template file. You can then share the template with other administrators or import the template to other NetScaler appliances that require a similar AppExpert application configuration.

To get started with an AppExpert application, you must first obtain the appropriate application template and import the template to the NetScaler appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components, statistics, and the Application Visualizer. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

AppExpert Application Terminology

Following are the terms used in the AppExpert applications feature and the descriptions of the entities for which the terms are used:

Public Endpoint. The IP address and port combination at which the NetScaler appliance receives client requests for the associated web application. A public endpoint can be configured to receive either HTTP or secure HTTP (HTTPS) traffic. All client requests for the web application must be sent to a public endpoint. An AppExpert application can be assigned multiple endpoints. You configure public endpoints after you import a template.

Application Unit. An AppExpert application entity that processes a subset of web application traffic and load balances a set of services that host the associated content. The subset of traffic that an application unit must manage is defined by a rule. Each application unit also defines its own set of traffic optimization and security policies for the requests and responses that it manages. The NetScaler services associated with these policies are Compression, Caching, Rewrite, Filter, Responder, and Application Firewall.

By default, every AppExpert application with at least one application unit includes a default application unit, which cannot be deleted. The default application unit is not associated with a rule for identifying requests and is always placed last in the order of application units. It defines a set of policies for processing any request that does not match the rules that are configured for the other application units, thereby ensuring that all client requests are processed.

Application units and their associated rules, policies, and actions are included in AppExpert application templates.

Service. The combination of the IP address of the server that hosts the web application instance and the port to which the application is mapped on the server, in the format <IP address>:<Port>. A web application that serves a large number of requests is usually hosted on multiple servers. Each server is said to host an instance of the web application, and each such instance of the web application is represented by a service on the NetScaler appliance. Services are deployment-specific, and are therefore not included in templates. You must configure services after you import a template.

Application Unit Rule. Either a classic expression or a default syntax expression that defines the characteristics of a traffic subset for an application unit. The following example rule is a default syntax expression that identifies a traffic subset that consists of four image types:

```
HTTP.REQ.URL.SUFFIX.EQ("bmp") || HTTP.REQ.URL.SUFFIX.EQ("gif") ||
HTTP.REQ.URL.SUFFIX.EQ("png") || HTTP.REQ.URL.SUFFIX.EQ("jpg")
```

For more information about default syntax expressions and classic policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX123868.

Traffic Subset. A set of client requests that require a common set of traffic optimization and security policies. A traffic subset is managed by an application unit and is defined by a rule.

How an AppExpert Application Works

When the endpoint receives a client request, the NetScaler appliance evaluates the request against the rule that is configured for the topmost application unit. If the request satisfies this rule, the request is processed by the policies that are configured for the application unit, and then forwarded to a service. The choice of service depends on which services are configured for the application, and on settings such as the load balancing algorithm and persistence method configured for the application unit.

If the request does not satisfy the rule, the request is evaluated against the rule for the next topmost application unit. In this order, the request is evaluated against each application unit rule until the request satisfies a rule. If the request does not satisfy any of the configured rules, it is processed by the default application unit, which is always the last application unit.

You can configure multiple public endpoints for an AppExpert application. In such a configuration, by default, each application unit processes requests received by all the public endpoints and load balances all the services that are configured for the application. However, you can specify that an application unit processes traffic from only a subset of the public endpoints and load balances only a subset of the services that are configured for the AppExpert application.

Getting Started with an AppExpert Application

The process of setting up an AppExpert application begins with downloading the appropriate AppExpert application template from the Citrix Community Web site at http://community.citrix.com/display/ns/AppExpert+Templates. The template that you need depends on the NetScaler release running on your appliance.

After you download the template, you must import the template to the NetScaler appliance, configure deployment settings, and then verify the configuration to make sure that the AppExpert application is working as expected.

Importing an AppExpert Application Template

You can either import the template file directly from your local computer or upload the template to the appliance and then import it. For more information about uploading a template to the NetScaler appliance, see Uploading and Downloading Template Files.

During import, along with the template file that you specify in the AppExpert Template Wizard, you can include a deployment file that contains deployment details. If you choose to include a deployment file, you do not have to provide any additional information. All application-configuration information is imported from the template file and all deployment-specific information for the application is imported from the deployment file. The NetScaler appliance imports all configuration settings from the deployment file through the NITRO API and the wizard displays the configuration summary screen for your verification. If you do not include a deployment file, the wizard displays screens on which you can specify deployment information. During import, if an error occurs, the configuration changes that were made during the import process are automatically rolled back. During import, if an error occurs, any changes are automatically rolled back, preserving the configuration that was in place before you attempted to import the AppExpert application. For more information about the format of application templates and deployment files, see Understanding NetScaler Application Templates and Deployment Files. For more information about how the template and deployment files are imported through the NITRO API, see the *Citrix NetScaler NITRO SDK for C# Getting Started Guide*, the *Citrix NetScaler NITRO Getting Started Guide for REST API*, and the *Citrix NetScaler NITRO SDK for Java Getting Started Guide*.

Note: The deployment file must contain information about only one public endpoint. Additionally, the application template file and the deployment file must be valid XML files. You cannot import a template file that is in GZIP format.

During import, you can configure deployment settings such as a public endpoint, services, and service groups. During import, you configure only one public endpoint, but you can specify as many services and service groups as you want the AppExpert application to manage. After you import the template, you can specify additional endpoints, services, and service groups for the configuration. If the public endpoint that you configure during import

uses the HTTPS protocol, you must also specify a server certificate for the public endpoint. Additionally, if variables have been configured for the template, a **Specify Variable Values** page appears in the wizard. On this page, you can choose to specify new values for the variables. For more information about configuring endpoints after you import a template, see Configuring Public Endpoints. For more information about configuring services and service groups after you import a template, see Configuring Services and Service Groups. For more information about configuring variables for a NetScaler application, see Creating Variables in Application Templates.

To import an AppExpert application template to the NetScaler appliance

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click **Applications**, and then click **Import**.
3. Follow the instructions in the **AppExpert Template Wizard**.

Verifying and Testing the Configuration

Verification is an important step in the process of setting up the NetScaler application. Before you proceed with other configuration tasks, you must verify that the state of the entities, such as endpoints and application units, are UP and then test the entities for correct processing.

Verifying the Configuration

The graphical user interface (GUI) includes icons that indicate the states of the entities in the AppExpert application. These icons are displayed for applications and application units and are based on the health checks that the NetScaler appliance performs periodically on services and entities. The following table lists the icons and describes their meanings.

Table 1. Descriptions of State Indictor Icons

Icon	Entity	Indicates that

	Application	At least one public endpoint is up. The application will accept client requests from the public endpoints that are up.
	Application unit	The application unit is up. The application unit is up when at least one service or service group is up.
	Application	The public endpoint is out of service (disabled). This indicator is displayed when only one public endpoint is configured for the AppExpert application.
	Application	All the endpoints that are configured for the application are out of service. This indicator is displayed only when multiple endpoints are configured for the application.
	Application unit	All the services configured for the application unit are down.

You must ensure that the icons for each application and its application units are green at all times. If the icon that is displayed for an application is not green, verify that you have configured the public endpoints correctly. If the icon that is displayed for an application unit is not green, verify that the services are configured correctly. However, note that a green indicator does not mean that the state of all associated entities is UP. This only means that the application has sufficient resources (endpoints and services) to serve client requests. To verify that the state of all associated entities is UP, check the health of all the entities on the statistics page for the application. For more information about viewing the application statistics page, see Viewing Application Statistics.

Testing the Configuration by Using Hit Counters

You can test the configuration by sending test HTTP requests for web application content through the NetScaler appliance, and then verifying that the requests are being processed by the right application units by viewing the hit counters for the various AppExpert application entities. For example, to verify that the endpoint is receiving requests, you view the hit counter for the AppExpert application. To verify that the configured application unit rules are being matched as expected, you view the hit counters for the AppExpert application units.

Note: To view hit counters for policies and actions that are configured for AppExpert applications, you must go to the associated feature node. For example, to view the hit counter for a Rewrite policy that is configured for an AppExpert application, you must go to the Rewrite feature node in the NetScaler configuration utility.

For a test example, consider an AppExpert application that includes an application unit called "WebPages" for processing web page content, and an application unit called "Images" for processing images. In this example, the rule that is configured for the application unit "WebPages" includes an expression that checks whether an HTML file is being requested. The "Images" application unit includes an expression that checks whether an image file is being requested.

Consider an HTML file called "sitehome.html," located at "/var/www/html/myapplicationpages/," on a backend server with an IP address of 192.0.2.10. In addition to HTML content, the HTML file also references images stored on the server. An HTTP request for the HTML file, sent directly to the server, would be as follows:

```
http://192.0.2.10/myapplicationpages/sitehome.html
```

To send a test request for this file through the NetScaler appliance, in the URL, replace the IP address of the server with the IP address of the public endpoint that is configured for the AppExpert application. For example, if the IP address of the public endpoint is 192.0.2.11, your test URL would be as follows:

```
http://192.0.2.11/myapplicationpages/sitehome.html
```

After you send the request, you must view the hit counter for the application to verify that the public endpoint received the request, view the hit counter for the application unit "webpages" to verify that the request for the HTML file matched the rule configured for the application unit, and view the hit counter for the application unit "images" to verify that the requests for the images matched the rule configured for the application unit.

For the application, the **Hits** dialog box displays the total number of requests received by each configured public endpoint. For an application unit, the **Hits** dialog box displays the

number of requests that the application unit processed from each of the public endpoints and the total hit count.

To view the hit counter for an application or application unit

1. In the navigation pane of the configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click the application or application unit for which you want to view the hit counter.
3. Click **Hits**.

Customizing the Configuration

After you verify that the AppExpert application is working correctly, you can customize the configuration to suit your requirements.

You can configure public endpoints and services for the AppExpert application and specify only a subset of the endpoints and services for each application unit. When you want the AppExpert application to manage a traffic subset that is not included in the template, you can either add an application unit for the new traffic subset or modify an existing application unit rule. You can also specify the order of evaluation of the traffic subsets that the AppExpert application manages.

Finally, you can modify the policies that the template provided. If the AppExpert application template does not include policies for a particular NetScaler feature, such as Rewrite or Application Firewall, you can configure your own policies.

The order in which you perform these tasks depends on your requirement. However, before you configure a service for an application, you must configure the service for the parent application.

Configuring Public Endpoints

When importing the AppExpert application template, the AppExpert Template Wizard allows you to specify only one public endpoint. After you import the AppExpert application template, you can specify additional public endpoints for the AppExpert application and then bind them to application units. A public endpoint can be configured to recieve either HTTP or secure HTTP (HTTPS) traffic.

If endpoints are already configured for the application, you can also dissociate endpoints from the AppExpert application and delete any endpoints that you no longer need. Note that when you dissociate a public endpoint from the AppExpert application, the endpoint is automatically unbound from the associated application unit, but it is not deleted from the system.

To configure public endpoints for an AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click **Configure Public Endpoints**.
3. In the **Choose Public Endpoints** dialog box for the application, do one of the following:
   - If the endpoints you want are listed in the dialog box, click the corresponding check boxes.
   - If you want to specify all the public endpoints, click **Activate All**.
   - If you want to dissociate endpoints from the AppExpert application, clear the corresponding check boxes.

     If you want to create a new public endpoint, click **Add**. Then, in the **Create public endpoint** dialog box, configure endpoint settings, and then click **OK**.

     In the **Create public endpoint** dialog box, you can specify only the name, IP address, port, and protocol for the endpoint. You can specify additional endpoint settings after you create the public endpoint. To specify additional endpoint settings, after you create the endpoint, in the **Choose Public Endpoints** dialog box, click the endpoint, and then click **Open**. Then, in the **Configure Public Endpoint** dialog box, provide additional settings, and then click **OK**.

     For more information about the parameters in the **Create public endpoint** and **Configure Public Endpoint** dialog boxes, see the "Content Switching" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

     If you want to modify a public endpoint, click the endpoint, and then click **Open**. Then, in the **Configure Public Endpoint** dialog box, modify settings for the endpoint, and then click **OK**.

     For more information about the parameters in the **Configure Public Endpoint** dialog box, see the "Content Switching" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.
4. Click **Close**.

Configuring Endpoints for an Application Unit

When you configure multiple public endpoints for an AppExpert application, by default, all endpoints are bound to each application unit, and each application unit processes the requests received by all the endpoints. However, you can specify that a given application unit manages the traffic that is received by only a subset of the endpoints that are configured for the AppExpert application.

To configure endpoints for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click **Configure Public Endpoints**.
3. In the **Choose Public Endpoints** dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click the corresponding check boxes.
4. Click **OK**.

Configuring Services and Service Groups

When you configure a service or service group, you either modify an existing service or service group, or add new services to the AppExpert application. You add services or service groups if you did not specify them when you imported the application template. You also add services and service groups when you increase the number of servers that host instances of the application. You can configure a service and service group for an application unit only after you configure the service or service group for the AppExpert application.

To configure a service or service group for the AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application for which you want to configure services or service groups, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. On the **Service** or **Service Groups** tab, do one of the following:
   - If the services or service groups that you want are listed on the tab, click the corresponding check boxes.
   - If you want to specify all the services or service groups, click **Activate All**.
   - If you want to create a new service or service group, click **Add**. Then, in the **Create Service** dialog box or **Create Service Group** dialog box, configure settings for the service or service group, respectively, and then click **Create**.
   - If you want to modify a service, click the service, and then click **Open**. Then, in the **Configure Service** dialog box or **Create Service Group** dialog box, configure settings for the service or service group, respectively, and then click **OK**.

For information about the settings in the **Create Service**, **Configure Service**, and **Create Service Group** dialog boxes, see the "Load Balancing" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

Configuring Services, Service Groups, and Load Balancing Parameters for an Application Unit

When you configure services and service groups for an AppExpert application, by default, all the services and service groups are bound to each application unit. However, depending on how you have configured your web application, the application resources that are managed by an application unit might be hosted on only some of the servers that are configured as services for the AppExpert application. Or, a set of servers might host content that is meant for the requests received at one or more specific public endpoints. In such scenarios, if all the services and service groups that are configured for the AppExpert application are associated with the application unit, a request that is forwarded to a server that does not host the requested content might not be served or might be served incorrect content. Therefore, you must ensure that each application unit is configured to manage only those services that can serve the requested content.

When configuring services and service groups for an application unit, you might choose to specify load balancing settings such as the weights that services must be assigned and the desired load balancing, persistence, and spillover methods. For more information about these settings, see the "Load Balancing" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

To configure services or service groups for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. In the **Services** or **Service Groups** tab, do one of the following:
   - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the **Weight** column, specify the weight that you want to assign to each configured service.
   - To specify all services or service groups, click **Activate All**.
5. On the **Method and Persistence** and **Advanced** tabs, specify the desired parameters.
6. Click **OK**.

Creating Application Units

You might need to add application units for traffic subsets that are either specific to your web application implementation or not defined in the template. When creating an application unit, you must configure a rule for the application unit.

To create an application unit for the AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Create Application Unit** dialog box, specify values for the following parameters:
   - **Name***—The name that you want to assign to the application unit.
   - **Rule***—The rule that identifies the traffic subset that the application unit will manage.
   - **Classic Syntax**—To specify that you want to configure a classic expression in the **Rule** box, click this option button.
   - **Default Syntax**—To specify that you want to configure a default syntax expression in the **Rule** box, click this option button.

   *A required parameter
4. Click **Create**.

Configuring Application Unit Rules

You might want to configure an application unit rule to include or exclude certain types of traffic. When you configure the rule, you can also define the syntax of the expression.

To configure an application unit rule

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to modify the rule, and then click **Open**.
3. In the **Configure Application Unit** dialog box, do the following:
   a. To specify the format of the new expression, do one of the following:
      - To specify that you want to configure a classic expression in the **Rule** box, click **Classic Syntax**.
      - To specify that you want to configure an advanced expression in the **Rule** box, click **Default Syntax**.
   b. In the **Rule** box, configure the expression.

      For more information, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX123868.
4. Click **OK**.

Specifying the Order of Evaluation of Application Units

Application unit rules are evaluated in the order in which they are placed in the graphical user interface (GUI). The rule that is configured for the topmost application unit is always configured first, followed by the rule that is configured for the second topmost application unit, and so on. The default application unit is always evaluated last.

When a request matches the rule that is configured for an application unit, the request is processed by the application unit, and no further matching is performed. Therefore, the order of evaluation of application units becomes an important factor if the traffic subsets for two or more application units overlap. If the traffic subsets for two or more application units overlap, you must specify the order in which an incoming request is matched against the application unit rules.

To specify the order of evaluation of application units

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, do the following:
   - To move an application unit up by one step, right-click the application unit, and then click **Move Up**.
   - To move an application unit down by one step, right-click the application unit, and then click **Move Down**.

Configuring Policies for Application Units

For an AppExpert application, you can configure policies for Compression, Caching, Rewrite, Filter, Responder, and Application Firewall. The templates that you download from the Citrix Community web site provide you with a set of policies that fulfill the most common application management requirements. You might want to fine-tune or customize these policies. If the set of policies provided for a given application unit does not include policies for a particular feature, you can create and bind your own policies for that feature.

If you create an AppExpert application without using a template, you must configure all the policies that the web application needs.

The GUI uses various icons to indicate whether or not policies are configured for a feature. For an application unit, if a policy is configured for a given feature, an icon that represents the feature is displayed. For example, if a compression policy is configured for an application unit, a compression icon is displayed in the **Compression** column for the application unit. For features for which no policy is configured, an icon depicting a plus sign (+) is displayed.

Note: When configuring policies for application units, you might need to configure policies and expressions that are either in the classic or default syntax. Additionally, when you configure default syntax policies, you might need to specify parameters such as Goto expressions and invoke policy banks. For information about configuring policies and expressions in both formats, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX123868.

Configuring Compression Policies

You can use either classic policies or advanced policies to configure compression, but you cannot bind compression policies of both types to the same application unit.

To configure a compression policy for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Compression** column.

3. In the **Configure Compression Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:

   - Click **Switch to Default Syntax** if you want to configure a default syntax compression policy. If you want to bind or configure classic compression policies, and if you are in the default syntax view, you can click **Switch to Classic Syntax** to return to the classic policy view and begin modifying bound classic policies or create and bind new classic compression policies.

     Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you are in the default syntax view, when you click **Insert Policy**, the list that appears in the **Policy Name** column will include only default syntax policies. You cannot bind policies of both types to an application unit.

   - If you want to configure classic policies, click either **Request** or **Response**, depending on whether you want the policy to be evaluated at request-time or at response-time.

     You can configure both request-time and response-time classic compression policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.

   - To modify a compression policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Compression Policy** dialog box, modify the policy, and then click **OK**.

     For information about modifying a compression policy, see the "Compression" chapter in the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX123870.

   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.

   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.

   - To regenerate assigned priorities, click **Regenerate Priorities**.

   - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Compression Policy** dialog box, configure the policy, and then click **Create**.

For more information about creating a compression policy, see the "Compression" chapter in the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX123870.

- If you are configuring a default syntax expression, do the following:

  - In the **Goto Expression** column, select a **Goto expression**.
  - In the **Invoke** column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click **Apply Changes**, and then click **Close**.

Configuring Caching Policies

You can use only default syntax policies and expressions to configure Caching policies.

To configure Caching policies for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Caching** column.

3. In the **Configure Cache Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:

   Click either **Request** or **Response**, depending on whether you want the policy to be evaluated at request-time or at response-time.

   You can configure both request-time and response-time Caching policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.

   To modify a Caching policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Cache Policy** dialog box, modify the policy, and then click **OK**.

   For information about modifying a Caching policy, see the "Integrated Caching" chapter in the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX123870.

   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.

   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.

   - To regenerate assigned priorities, click **Regenerate Priorities**.

     To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Cache Policy** dialog box, configure the policy, and then click **Create**.

     For information about creating a Caching policy, see the "Integrated Caching" chapter in the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX123870.

   - In the **Goto Expression** column, select a **Goto expression**.

   - In the **Invoke** column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Rewrite Policies

You can use only default syntax policies and expressions to configure Rewrite policies.

To configure Rewrite policies for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Rewrite** column.
3. In the **Configure Rewrite Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:

   Click either **Request** or **Response**, depending on whether you want the policy to be evaluated at request-time or at response-time.

   You can configure both request-time and response-time Rewrite policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.

   To modify a Rewrite policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Rewrite Policy** dialog box, modify the policy, and then click **OK**.

   For information about modifying a Rewrite policy, see Rewrite.

   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.

     To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Rewrite Policy** dialog box, configure the policy, and then click **Create**.

     For information about creating a Rewrite policy, see Rewrite.

   - In the **Goto Expression** column, select a **Goto expression**.
   - In the **Invoke** column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Filter Policies

You can use only classic policies and expressions to configure Filter policies.

To configure Filter policies for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Filter** column.
3. In the **Configure Filter Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:

   To modify a Filter policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Filter Policy** dialog box, modify the policy, and then click **OK**.

   For information about modifying a Filter policy, see Content Filtering.

   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.

   To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Filter Policy** dialog box, configure the policy, and then click **Create**.

   For information about creating a Filter policy, see Content Filtering.
4. Click **Apply Changes**, and then click **Close**.

Configuring Responder Policies

You can use only default syntax policies and expressions to configure Responder policies.

To configure Responder policies for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Responder** column.
3. In the **Configure Responder Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:

   To modify a Filter policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Responder Policy** dialog box, modify the policy, and then click **OK**.

   For information about modifying a Responder policy, see Responder.

   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.

     To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Responder Policy** dialog box, configure the policy, and then click **Create**.

     For information about creating a Responder policy, see Responder.

   - In the **Goto Expression** column, select a **Goto expression**.
   - In the **Invoke** column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Application Firewall Policies

You can configure both classic and default syntax policies and expressions for Application Firewall. However, if a policy of one type is already bound globally or to a virtual server that is configured on the appliance, you cannot bind a policy of the other type to an application unit. For example, if a default syntax policy is already bound either globally or to a virtual server, you cannot bind a classic policy to an application unit.

To configure Application Firewall policies for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Application Firewall** column.

3. In the **Configure Application Firewall Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:

   Click either **Classic Expression** or **Advanced Expression** depending on the type of expression you want to configure for the Application Firewall policy.

   Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you select **Advanced Expression**, when you click **Insert Policy**, the list that appears in the **Policy Name** column will include only default syntax policies. You cannot bind policies of both types to an application unit. This option is not available if a policy of either type is already bound either globally or to a virtual server.

   To modify a compression policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Application Firewall Policy** dialog box, modify the policy, and then click **OK**.

   For information about modifying a Application Firewall policy, see the "Policies" chapter in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX123855.

   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.

   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.

   - To regenerate assigned priorities, click **Regenerate Priorities**.

     To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Application Firewall Policy** dialog box, configure the policy, and then click **Create**.

     For information about creating an Application Firewall policy, see the "Policies" chapter in the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX123855.

4. Click **Apply Changes**, and then click **Close**.

Viewing AppExpert Applications and Configuring Entities by Using the Application Visualizer

The Application Visualizer is a graphical representation of an AppExpert application. The Visualizer displays the public endpoints, application units, backend services, and policies that are configured for the application. You can use the Visualizer to obtain a visual overview of an AppExpert application's configuration and configure some of the displayed entities. By default, the Visualizer displays application units, services, and monitors for the selected application.

On NetScaler 9.3 nCore, you can also use the Application Visualizer to monitor some AppExpert application parameters. For more information about monitoring application parameters by using the Application Visualizer, see "Monitoring an Application by Using the Application Visualizer".

To view an AppExpert application by using the Application Visualizer

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, click the name of the application that you want to view, and then click **Visualizer**.

3. Do one or more of the following:

   - To optimize the display area, choose **Best Fit**, **Zoom In**, or **Zoom out**. If an item that you want to see disappears from view after zooming in, you can click and drag the viewable area.
   - To save the graph as an image file, click **Save Image**.
   - To find a particular entity, in the **Search in** field, type an entity name. In the view area, the entity names that begin with the search string are highlighted. To restrict the search, click the drop-down menu and select the specific entity that you want to search for.
   - To view the policies that an application uses, click one or more icons to display feature-specific policies. The policy types are Compression, Filter, Rewrite, Responder, Cache, Application Firewall, Authorization, Auditing, HTML Injection, SureConnect, Priority Queuing, and Traffic.
   - To view the rule that is configured for an application unit, click the curve that connects the public endpoint to the application unit. The rule is displayed on the **Related Tasks** tab.
   - To view the binding information for an application unit, policy, or monitor, click the displayed icon, click the **Related Tasks** tab, and then click **Show Bindings**
   - To view member services, click the icon for the service, click the **Related Tasks** tab, and then click **Show Member Services**.
   - To view detailed statistics for a public endpoint or application unit, click the icon that is displayed, click the **Related Tasks** tab, and then click **Statistics**.
   - To view the **Load Balancing Visualizer** for an application unit, click the application unit, click **Related Tasks**, and then click **Visualizer**.

To configure and view entities in an AppExpert application by using the Application Visualizer

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, click the name of the application that you want to configure or view, and then click **Visualizer**.

3. Do one or more of the following:

   - To configure an entity that is displayed in the viewing area, click the icon for the entity, click the **Related Tasks** tab, and then click **Modify public endpoint**.

     In the Application Visualizer, you can modify only public endpoints and services.
   - To bind additional monitors to a service, click the **Available Resources** tab, select **Monitors** from the drop-down list, and then click and drag a monitor to a service.
   - To unbind a service from an application unit, click the curve that connects the application unit and the service, click **Related Tasks**, and then click **Unbind**.
   - To unbind a monitor from a service, click the curve that connects the service and the monitor, click **Related Tasks**, and then click **Unbind**.
   - To modify a monitor, click the monitor, click **Related Tasks**, and then click **Open**.
   - To modify the binding parameters for a monitor, click the curve that connects the monitor to the associated service, click **Related Tasks**, and then click **Modify Parameters**.
   - To apply a common service configuration across multiple service containers that are displayed when the services bound to a vserver do not have the same configuration, click the service container whose configuration you want to apply to all the containers, and then, in **Related Tasks**, click **Apply Configuration**.
   - To view a comparative list of the parameters whose values differ across service containers, click the icon for a container, click the **Related Tasks** tab, and then click **Service Attributes Diff**. The comparative list helps you determine which service container has the service configuration that you want to apply to all the containers. After you determine which service container has the configuration you want, right-click the container, and then click **Apply this Configuration**.
   - To copy the configuration of an entity (other than the configuration of the AppExpert application) to the local computer's clipboard, click the entity, click the **Related Tasks** tab, and then click **Copy Properties**. You can then paste the configuration information in a word processing document or spreadsheet.

Monitoring a NetScaler Application

After you customize the AppExpert application, you can view application statistics to make sure that the application and all its entities are working correctly. On NetScaler 9.3 nCore, you can also use the Application Visualizer to monitor statistics associated with certain entities such as policies and virtual servers.

You can also view the hit counters for various entities at regular intervals to make sure that counters are being updated.

Viewing Application Statistics

In the **Applications** node, you can select an application and view the **Statistics** page for the application. On the **Statistics** page, you can monitor the health and states of public endpoints and application units, and view the following statistical information:

- Requests and responses per second for each of the public endpoints and application units.
- Bytes per second, at each endpoint, for incoming and outgoing traffic.
- Application unit hit counters and the number of client and server connections for each application unit.
- Statistics for the services that are bound to the application units.

On the **Statistics** page, you can also view CPU usage, memory usage, and system logs.

To view statistics for an application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click the application for which you want to view statistics, and then click **Statistics**.

Monitoring an Application by Using the Application Visualizer

On NetScaler 9.3 nCore, you can use the Application Visualizer to monitor the number of requests received per second at a given point in time by the vservers and the number of hits per second at a given point in time for Rewrite, Responder, and Cache policies.

To view statistical information for vservers, Rewrite policies, Responder policies, and Cache policies in the Visualizer

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, select the application for which you want to view statistical information, and then click **Visualizer**.
3. In the **Application Visualizer** window, do the following:

   To view the statistics, click **Show Stats**.

   The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time and has to be refreshed manually.

   - To refresh the statistical information, click **Refresh Stats**.

Viewing Hits

The hit counters that are provided for various AppExpert application entities enable you to monitor the functioning of public endpoints and application units. For an application, the Hits dialog box displays the total number of requests received by each configured public endpoint. For an application unit, the Hits dialog box displays the number of requests that the application unit processed from each of the public endpoints and the total hit count. For instructions on viewing hit counters, see Verifying and Testing the Configuration.

Deleting an Application

If you no longer need an application and its application units, you can delete it. When you delete an AppExpert application, backend services are not deleted, and any public endpoints that the application used become available for use by other applications.

When deleting an application, you are also prompted to specify whether you want to delete any bound policies and actions that are not used elsewhere.

To delete an application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click the name of the application that you want to delete, and then click **Remove**.

Configuring Authentication, Authorization, and Auditing

You can configure Authentication, Authorization, and Auditing (AAA) for the applications that you configure on the appliance. An authentication policy that is configured for an application defines the type of authentication to apply when a user or group attempts to access the application. If external authentication is used, the policy also specifies the external authentication server. Authorization policies configured for an application specify whether a particular user or group can access the application. Auditing policies define the audit log type, the level at which logging is performed, and other audit server settings. Authentication and auditing policies use the classic policy format.

Authentication policies, authorization policies, and auditing policies can be configured in any order. However, before you configure AAA for an application, you must configure a public endpoint for the application.

Configuring Authentication

Configuring authentication for an application involves specifying an authentication FQDN, an authentication virtual server, a server certificate, and authentication and session policies. Authentication policies are automatically bound to the authentication virtual server specified for the application.

To configure authentication for an AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click the name of the application for which you want to configure authentication, and then click **Authentication**.
3. In the **Authentication Wizard**, on the **Introduction** page, click **Next**.
4. Follow the instructions in the **Authentication Wizard**.

Configuring Authorization

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure permissions for a AAA user or group to access an AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click the AppExpert application for which you want to configure user or group access, and then click **Authorization**.
3. Do one of the following:
   - If the AAA user or group for which you want to configure permissions is already in the **Groups/Users** tree, drag the user or group from the **Groups/Users** tree to the **Users** or **Groups** node in the application tree. Then, right-click the user or group and click **Allow**.

     If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the application tree, right-click **Users** or **Groups**, and then click **Add**. In the **Create AAA Group** or **Create AAA User** dialog box, fill in the values, click **Create**, and then click **Close**.

     The user or group is created with the permission set to **Allow**. To change the permission setting, right-click the group or user, and then click the permission setting.
4. Click **Close**.

Configuring Auditing

When you configure auditing policies for an application, you must specify the server to which the log messages must be directed, the format of the messages logged, and the log level. Optionally, you can configure other settings, such as the log facility and date format. Auditing policies are automatically bound to all the AppExpert application’s public endpoints.

To configure auditing policies for an application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, click the application for which you want to configure auditing policies, and then click **Auditing**.

3. In the **Configure Auditing Policies** dialog box, click **Insert Policy**.

   To specify an existing auditing policy, under **Policy Name**, click the name of the policy, and then do the following:

   - To modify the priority that is assigned to the policy by default, under **Priority**, double-click the priority, and then type a new priority value.

   - To modify the settings of the audit server, under **Server**, double-click the name of the server, and then, in the **Configure Auditing Server** dialog box, modify the settings as appropriate. You can modify all the settings in this dialog box except the name of the audit server and the audit type. For more information about the settings in the **Configure Auditing Server** dialog box, see Auditing Policies.

   To create a new auditing policy, under **Policy Name**, click **New Policy**, and then, in the **Create Auditing Policy** dialog box, do the following:

   - In the **Name** box, type a name for the policy.

   - The **Name** box already contains the string that is required at the beginning of the server name. You cannot modify the string.

   - From the **Auditing Type** list, select the auditing type (either **SYSLOG** or **NSLOG**).

   - If the audit server you want to specify is already listed in the **Server** list, select the server from the list, and then, if you want to modify the server settings, click **Modify**. In the **Configure Auditing Server** dialog box, modify the settings as appropriate, and then click **OK**. For more information about the settings in the **Configure Auditing Server** dialog box, see Auditing Policies.

   - If you want to configure a new audit server, click **New**, and then, in the **Create Auditing Server** dialog box, type a name for the server, specify the server IP address, port number, and other settings as appropriate. When finished, click **OK**.

   - Click **Create**.

   - To change the priorities for the new auditing policies you created, under **Priority**, for each policy for which you want to change the priority, double-click the priority value and type new priority value.

   - To regenerate priorities, click **Regenerate Priorities**.

   - To unbind a policy, click the policy, and then click **Unbind Policy**.

   - To modify a policy, click the policy, and then click **Modify Policy**.

4. Click **Apply Changes**, and then click **Close**.

Disabling AAA for an Application

After you configure AAA for an application, you can disable the AAA configuration for that application. When you disable AAA for an application, the configuration is not lost. You can enable AAA for the application when you want to reapply the configuration.

To enable or disable AAA for an application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, click the application for which you want to enable or disable AAA, and then do one of the following:
   - To disable AAA for the application, click **Turn Off AAA**.
   - To enable AAA for the application, click **Turn On AAA**.

Setting Up a Custom NetScaler Application

If an AppExpert application template is not available for the Web application that you want to manage through the NetScaler appliance, or if available AppExpert application templates do not suit your requirements, you can create an AppExpert application without a template.

To create an AppExpert application without a template, you must first create an application and application units. Then, you configure public endpoints, services, and service groups. Finally, you configure the policies that determine how application traffic is evaluated and processed.

After you create the application and application units and configure policies, you must verify the configuration and test it to make sure that it is working correctly, just as you would when you configure an application by using a prebuilt AppExpert application template. Then, you must monitor the application to make sure that the application and its entities are working correctly.

Creating an Application

When you create an AppExpert application, the appliance creates a container to which you can add application units. The *default* application unit is not created until you create the first application unit.

To create an AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click **Applications**, and then click **Add**.
3. In the **Create Application** dialog box, in **Name**, enter a name for the application, and then click **OK**.

Creating Application Units

For each subset of traffic associated with your web application, you must create an application unit.

To create an application unit for the AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Create Application Unit** dialog box, specify values for the following parameters:
   - **Name***—The name that you want to assign to the application unit.
   - **Rule***—The rule that identifies the traffic subset that the application unit will manage.
   - **Classic Syntax**—To specify that you want to configure a classic expression in the **Rule** box, click this option button.
   - **Default Syntax**—To specify that you want to configure a default syntax expression in the **Rule** box, click this option button.

   *A required parameter
4. Click **Create**.

Configuring Public Endpoints for an AppExpert Application

After you have created all the application units that you require, you must configure one or more public endpoints to enable clients to access the web application through the NetScaler appliance.

To configure public endpoints for an AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click **Configure Public Endpoints**.
3. In the **Choose Public Endpoints** dialog box for the application, do one of the following:
   - If the endpoints you want are listed in the dialog box, click the corresponding check boxes.
   - If you want to specify all the public endpoints, click **Activate All**.
   - If you want to dissociate endpoints from the AppExpert application, clear the corresponding check boxes.

     If you want to create a new public endpoint, click **Add**. Then, in the **Create public endpoint** dialog box, configure endpoint settings, and then click **OK**.

     In the **Create public endpoint** dialog box, you can specify only the name, IP address, port, and protocol for the endpoint. You can specify additional endpoint settings after you create the public endpoint. To specify additional endpoint settings, after you create the endpoint, in the **Choose Public Endpoints** dialog box, click the endpoint, and then click **Open**. Then, in the **Configure Public Endpoint** dialog box, provide additional settings, and then click **OK**.

     For more information about the parameters in the **Create public endpoint** and **Configure Public Endpoint** dialog boxes, see the "Content Switching" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

     If you want to modify a public endpoint, click the endpoint, and then click **Open**. Then, in the **Configure Public Endpoint** dialog box, modify settings for the endpoint, and then click **OK**.

     For more information about the parameters in the **Configure Public Endpoint** dialog box, see the "Content Switching" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.
4. Click **Close**.

Configuring Public Endpoints for an Application Unit

For an application unit, you specify public endpoints in the same way as you would specify public endpoints for an application that is created from an AppExpert application template. For more information about specifying a subset of the endpoints for an application unit, see "Configuring Endpoints for an Application Unit".

To configure endpoints for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click **Configure Public Endpoints**.
3. In the **Choose Public Endpoints** dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click the corresponding check boxes.
4. Click **OK**.

Configuring Services and Service Groups for an AppExpert Application

Services and service groups are available for application units only after you configure the services and service groups for the AppExpert application. Therefore, you must configure services and service groups for the AppExpert application before you configure the services for the application units. All the services and service groups that you configure for an AppExpert application must use the same protocol (either HTTP or HTTPS). The procedure for configuring services and service groups for an AppExpert application that is not created from a template is the same as that for an application created from a template.

To configure a service or service group for the AppExpert application

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application for which you want to configure services or service groups, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. On the **Service** or **Service Groups** tab, do one of the following:
   - If the services or service groups that you want are listed on the tab, click the corresponding check boxes.
   - If you want to specify all the services or service groups, click **Activate All**.
   - If you want to create a new service or service group, click **Add**. Then, in the **Create Service** dialog box or **Create Service Group** dialog box, configure settings for the service or service group, respectively, and then click **Create**.
   - If you want to modify a service, click the service, and then click **Open**. Then, in the **Configure Service** dialog box or **Create Service Group** dialog box, configure settings for the service or service group, respectively, and then click **OK**.

For information about the settings in the **Create Service**, **Configure Service**, and **Create Service Group** dialog boxes, see the "Load Balancing" chapter in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

Configuring Services and Service Groups for an Application Unit

After you configure services and service groups, you must configure services and service groups for each application unit. However, this step is not necessary if each backend service hosts all the content associated with the web application. You configure services and service groups for an application unit if the content associated with the application unit is hosted on only a subset of the backend servers.

To configure services or service groups for an application unit

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. In the **Services** or **Service Groups** tab, do one of the following:
   - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the **Weight** column, specify the weight that you want to assign to each configured service.
   - To specify all services or service groups, click **Activate All**.
5. On the **Method and Persistence** and **Advanced** tabs, specify the desired parameters.
6. Click **OK**.

Configuring Policies

The procedures for configuring policies for an AppExpert application that is created without using a template are the same as those for an AppExpert application that was created from a template. For more information, see "Configuring Policies for Application Units".

Creating and Managing Template Files

After you set up an AppExpert application and customize it to suit your requirements, you can create a template from the application and then share the template with other administrators. Or, you can create a template and then import the template to other NetScaler appliances that require a similar AppExpert application configuration. This simplifies and expedites the process of setting up similar applications on other appliances.You can also export a content switching configuration to a template file. When creating a template file, you can configure variables in the policy expressions and actions that are configured for an application.

AppExpert application template files can be exported either to the template directory on the NetScaler appliance or to a folder on your local computer. You can then upload and download the templates to and from the NetScaler appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

Exporting an AppExpert Application to a Template File

When you export an AppExpert application, all application-configuration information is exported to a template file and all deployment-specific information is exported to a deployment file. The string "_deployment" is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the NetScaler appliance, the template file is stored in the /nsconfig/nstemplates/applications directory on the NetScaler appliance and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory. For more information about the format of application templates and deployment files, see "Understanding NetScaler Application Templates and Deployment Files". If you have configured Access Gateway policies for the NetScaler application, when you export the AppExpert configuration, you can choose to include the Access Gateway policies in the template. For information about configuring Access Gateway policies for an AppExpert application, see "Access Gateway Applications".

To export an AppExpert application to a template file

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.

2. In the details pane, click the name of the application that you want to export as a template file, and then click **Export**.

3. In the **Export...as Template** dialog box, do the following:

   a. In the **Name** box, modify the name of the template, if necessary.

   b. If you want to configure variables for the template, click **Configure Variables**, and then, in the **Configure Variables** dialog box, configure the variables that you want.

      For more information about configuring variables in application templates, see "Creating Variables in Application Templates".

   c. If you want to export the template file to the application templates directory on the appliance, make sure that **Browse (Appliance)** is displayed.

   d. If you want to export the template file to your computer, click the **Browse (Appliance)** drop-down menu, click **Local**, browse to the location to which you want to save the file, and then click **Save**.

   e. Provide the following information:

      - **Introduction Description**—Any text that introduces the AppExpert application template during import. This text is displayed on the **Specify Application Name** page of the **AppExpert Template Wizard** when the template is imported.

      - **Summary Description**—Any summary that you might want to display on the **Summary** page of the **AppExpert Template Wizard** when the template is imported.

      - **Author**—The name of the author of the template.

      - **Major**—The major version number of the template.

      - **Minor**—The minor version number of the template. This number is appended to the major version number and displayed on the **Summary** page of the **AppExpert Template Wizard**, during import, in the format Major.Minor.

   f. Click **OK**.

      If Access Gateway policies have been configured for the application, you will be prompted to include the Access Gateway configuration in the application template. If you want to include the Access Gateway configuration in the template, at the prompt, click **Yes**.

Exporting a Content Switching Virtual Server Configuration to a Template File

You can also export a content switching configuration as an application template. You can export a content switching virtual server configuration to an application template either from the Content Switching Virtual Servers pane or from the Content Switching Visualizer. Configuration information, which includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies, is exported to a template file and all deployment-specific information is exported to a deployment file. The string "_deployment" is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the NetScaler appliance, the template file is stored in the /nsconfig/nstemplates/applications directory on the NetScaler appliance and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory. For more information about the format of application templates and deployment files, see "Understanding NetScaler Application Templates and Deployment Files".The configuration information that is exported includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies.

However, if the content switching virtual server is already configured as the public endpoint for an AppExpert application, you cannot export the configuration to a template file. In this scenario, you must export the associated AppExpert application to a template. For more information about exporting an AppExpert application to a template file, see "Exporting an AppExpert Application to a Template File".

To export a content switching configuration to an application template file from the Content Switching Virtual Servers pane

1. In the navigation pane of the NetScaler configuration utility, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click **Create AppExpert Template**.
3. Perform steps 4 through 6 described in To export a content switching configuration to an application template file from the Content Switching Visualizer.

To export a content switching configuration to an application template file from the Content Switching Visualizer

1. In the navigation pane of the NetScaler configuration utility, expand **Content Switching**, and then click **Virtual Servers**.

2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click **Visualizer**.

3. In the **Content Switching Visualizer**, click the icon for the content switching vserver, click **Related Tasks**, and then click **Create Template**.

4. In the **Export...as Template** dialog box, enter a name for the template file, and then do one of the following:

   - To export the template file to the appliance, make sure that **Browse (Appliance)** is displayed.

   - To export the template file to your computer, click the **Browse (Appliance)** drop-down menu, click **Local**, browse to the location to which you want to save the file, and then click **Save**.
5. Provide the following information:

   - **Introduction Description**—Any text that introduces the AppExpert application template during import. This text is displayed on the **Specify Application Name** page of the **AppExpert Template Wizard** when the template is imported.

   - **Summary Description**—Any summary that you might want to display on the **Summary** page of the **AppExpert Template Wizard** when the template is imported.

   - **Author**—The name of the author of the template.

   - **Major**—The major version number of the template.

   - **Minor**—The minor version number of the template. This number is appended to the major version number and displayed on the **Summary** page of the **AppExpert Template Wizard**, during import, in the format `Major.Minor`.
6. Click **OK**.

Creating Variables in Application Templates

Application templates support the declaration of variables in the policy expressions and actions that are configured for an application. The ability to declare variables in policy expressions and actions enables you to replace preconfigured values in expressions (for example, configurable parameters such as the host name of a server or the target for a Rewrite action) with values that suit the environment into which you are importing the template. If variables have been configured for an AppExpert application template, the AppExpert Template Wizard, which appears when you import an AppExpert application template, includes a **Specify Variable Values** page on which you can specify appropriate values for the variables that are configured for the template.

As an example, consider the following policy expression that is configured to evaluate the value of the Host header in an HTTP request:

```
HTTP.REQ.HEADER("Host").CONTAINS("server1")
```

If you want the server name to be configurable at import time, you can specify the string "server1" as a variable. When importing the template, you can specify a new value for the variable on the **Variables** tab.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

 In the export application template wizard, you can define variables in certain fields (fields with an adjacent button) for the following entities:

- Cache policies
- Rewrite policies
- Rewrite actions
- Responder policies
- Responder actions

To configure a variable in a policy expression or action

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application that you want to export to a template file, and then click **Export**.
3. In the **Export...as Template** dialog box, modify the default template file name if required, specify the location where you want to save the template, and then click **Configure Variables**.
4. In the **Configure Variables** dialog box, click the tab that lists the policy expression or action for which you want to configure a variable, select the expression, and then click **Configure Variables**.
5.  In the **Variables** dialog box, click the button next to the expression or value in which you want to create a variable.
6. In the **Variables** dialog box, do the following:

   To create a variable, in the text box that displays the configured expression or value, select the string that you want to be configurable at import time, and then click **Add**. In the **Add Variable** dialog box, specify a name and a description for the variable, and then click **Create**.

   - The name of the variable, its value, and the description you provided appear in the **Available Variables** listing in the dialog box. The name you provide will be the name of the associated field in the template import wizard, and the description will appear as alt text when the user positions the mouse pointer over the field.
   - To modify a variable, in the **Available Variables** list, click the variable, and then click **Open**. In the **Add Variable** dialog box, modify the value and the description, and then click **OK**.
   - To view all the strings that are assigned to a given variable, in the **Available Variables** listing, click the name of the variable. The strings that are assigned to the variable are highlighted.
   - To view a list of all the entities and parameters in which the variable is used, in the **Available Variables** listing, click the variable whose references you want to view, and then click **Show References**.

     To assign a string to an existing variable, in the text box that displays the expression you configured, select the string you want to assign to an existing variable, right-click the selection, click **Use Existing Selection**, and then click the name of the variable to which you want to assign the string.

     If a variable has multiple strings assigned to it, when you specify a new value for the variable during import, all strings assigned to the variable are replaced with the new value.
7. Click **Close**.

Uploading and Downloading Template Files

Template files can be uploaded from your local computer to the NetScaler appliance or downloaded from the appliance to your local computer. On the appliance, AppExpert application templates are always stored in the AppExpert application templates directory, which is /nsconfig/nstemplates/applications/.

To upload an AppExpert application template from your local computer to the NetScaler appliance

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Templates**.
2. In the details pane, click **Manage Templates**.
3. In the **Manage Application Templates** dialog box, click **Application Templates**, and then click **Upload**.
4. In the **Upload Application Template** dialog box, browse to the directory in which the template file is stored, click the template file, and then click **Select**.

   The template file is uploaded to the AppExpert application template directory on the appliance.

To download an AppExpert application template from the NetScaler appliance to your local computer

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Templates**.
2. In the details pane, click **Manage Templates**.
3. In the **Manage Application Templates** dialog box, click the **AppExpert** application template that you want to download, and click **Download**.
4. In the **Download Application Template** dialog box, browse to the location to which you want to save the file, and then click **Save**.

Renaming an Application Template

You can rename an application template that is stored in the AppExpert application templates folder on the appliance.

To rename an AppExpert applications template

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Templates**.
2. In the details pane, click **Manage Templates**.
3. In the **Manage Application Templates** dialog box, click the AppExpert application template that you want to rename, and then click **Rename**.
4. Enter a new name for the template, and then click **Close**.

Deleting an AppExpert Application Template

You can delete an application template that you no longer need.

To delete an AppExpert application template

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Templates**.
2. In the details pane, click the template that you want to delete, and then click **Remove**.

Understanding NetScaler Application Templates and Deployment Files

When you export a NetScaler application, the following two files are automatically created:

- **NetScaler application template file.** Contains application-configuration information such as application units, rules, and configured policies.
- **Deployment file.** Contains deployment-specific information such as public endpoints, services, associated IP addresses, and configured variables.

In the application template and deployment file, each unit of application-configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, each public endpoint and associated endpoint details are encapsulated within the `<appendpoint>` and `</appendpoint>` tags, and all the endpoint elements are encapsulated within the `<appendpoint_list>` and `</appendpoint_list>` tags.

Note: After you export a NetScaler application, you can add elements, remove elements, and modify existing elements before importing the application to a NetScaler appliance.

Example of a NetScaler Application Template

Following is an example of a template file that was created from a NetScaler application called "SharePoint_Team_Site":

```
<?xml version="1.0" encoding="UTF-8" ?>
<template>
<template_info>
  <application_name>SharePoint_Team_Site</application_name>
  <templateversion_major>1</templateversion_major>
  <templateversion_minor>1</templateversion_minor>
  <author>Ed</author>
  <introduction>An application for managing a SharePoint team site with images, reports, and, XML content.
  <summary>This template includes variables</summary>
  <version_major>9</version_major>
  <version_minor>3</version_minor>
  <build_number>38</build_number>
</template_info>
<apptemplate>
  <rewrite>
    <rewriteaction_list>
      <rewriteaction>
        <name>Rw_name</name>
        <type>replace</type>
        <target>HTTP.REQ.BODY(10000).AFTER_REGEX(re/number/).BEFORE_REGEX(re/address/)</target>
        <stringbuilderexpr>"NA"</stringbuilderexpr>
```

```
            <allow_unsafe_pi1>NO</allow_unsafe_pi1>
          </rewriteaction>
          <rewriteaction>
            .
            .
            .
          </rewriteaction>
          .
          .
          .
        </rewriteaction_list>
        <rewritepolicy_list>
          <rewritepolicy>
            <name>Rw_number_NA</name>
            <rule>HTTP.REQ.BODY(100000).CONTAINS("admin")</rule>
            <action>Rw_name</action>
          </rewritepolicy>
          <rewritepolicy>
            .
            .
            .
          </rewritepolicy>
          .
          .
          .
        </rewritepolicy_list>
      </rewrite>
      <appunit_list>
        <appunit>
          <name>SharePoint_Team_Sitedefault</name>
          <rule />
          <expressiontype>PE</expressiontype>
          <servicetype>HTTP</servicetype>
          <ipv46>0.0.0.0</ipv46>
          <ipmask>*</ipmask>
          <port>0</port>
          <range>1</range>
          <persistencetype>NONE</persistencetype>
          <timeout>2</timeout>
          <persistencebackup>NONE</persistencebackup>
          <backuppersistencetimeout>2</backuppersistencetimeout>
          <lbmethod>LEASTCONNECTION</lbmethod>
          <persistmask>255.255.255.255</persistmask>
          <v6persistmasklen>128</v6persistmasklen>
          <pq>OFF</pq>
          <sc>OFF</sc>
          <m>IP</m>
          <datalength>0</datalength>
          <dataoffset>0</dataoffset>
          <sessionless>DISABLED</sessionless>
          <state>ENABLED</state>
          <connfailover>DISABLED</connfailover>
          <clttimeout>180</clttimeout>
          <somethod>NONE</somethod>
          <sopersistence>DISABLED</sopersistence>
          <redirectportrewrite>DISABLED</redirectportrewrite>
```

```
        <downstateflush>DISABLED</downstateflush>
        <gt2gb>DISABLED</gt2gb>
        <ipmapping>0.0.0.0</ipmapping>
        <disableprimaryondown>DISABLED</disableprimaryondown>
        <insertvserveripport>OFF</insertvserveripport>
        <authentication>OFF</authentication>
        <authn401>OFF</authn401>
        <push>DISABLED</push>
        <pushlabel>none</pushlabel>
        <l2conn>OFF</l2conn>
      </appunit>
      <appunit>
        .
        .
        .
      </appunit>
      .
      .
      .
    </appunit_list>
    </apptemplate>
    <parameters>
      <property_list>
        <property>
          <variable_definition_list>
            <variable_definition>
              <name>body_size</name>
              <defaultvalue>10000</defaultvalue>
              <description>Evaluation Scope</description>
              <startindex>14</startindex>
              <length>5</length>
            </variable_definition>
            .
            .
            .
          </variable_definition_list>
          <object_type>rewriteaction</object_type>
          <object_name>Rw_name</object_name>
          <name>target</name>
        </property>
        .
        .
        .
      </property_list>
    </parameters>
</template>
```

Example of a Deployment File

Following is the deployment file associated with the "SharePoint_Team_Site" application in the preceding example:

```
<?xml version="1.0" encoding="UTF8" ?>
<template_deployment>
  <template_info>
    <application_name>SharePoint_Team_Site</application_name>
    <templateversion_major>1</templateversion_major>
    <templateversion_minor>1</templateversion_minor>
    <author>Ed</author>
    <introduction>An application for managing a SharePoint team site with images, reports, and, XML conter
    <summary>This template includes variables</summary>
    <version_major>9</version_major>
    <version_minor>3</version_minor>
    <build_number>38</build_number>
  </template_info>
  <appendpoint_list>
    <appendpoint>
      <ipv46>10.111.111.1</ipv46>
      <port>80</port>
      <servicetype>HTTP</servicetype>
    </appendpoint>
  </appendpoint_list>
  <service_list>
    <service>
      <ip>10.102.29.5</ip>
      <port>80</port>
      <servicetype>HTTP</servicetype>
    </service>
    <service>
      .
      .
      .
    </service>
    .
    .
    .
  </service_list>
  <variable_list>
    <variable>
      <name>body_size</name>
      <description>Evaluation Scope</description>
      <value>10000</value>
    </variable>
    <variable>
      .
      .
      .
    </variable>
    .
    .
    .
  </variable_list>
</template_deployment>
```

Access Gateway Applications

When you configure an AppExpert application to manage a web application through the Citrix® NetScaler® appliance, you also create a set of application units and configure a set of traffic optimization and security policies for each unit. The policies that you configure for each application unit (policies for features such as Compression, Caching, and Rewrite) evaluate traffic that is meant only for that unit. In addition to these policies, you might want to configure Access Gateway policies for the application as a whole to optimize the application traffic when accessed through the Access Gateway. The Access Gateway Applications feature enables you to configure Access Gateway policies (Authorization, Traffic, Clientless Access, and TCP Compression) for an AppExpert application. After you configure Access Gateway policies for AppExpert applications, you can include the policy configuration in the AppExpert application templates that you create.

You can also configure Access Gateway policies for intranet subnets, file shares, and other network resources.

Finally, you can create bookmarks for AppExpert applications and certain resources if you want users to be able to access them from the Access Gateway home page.

You can configure the entities in the Access Gateway Applications feature only by using the configuration utility.

How an Access Gateway Application Works

When you create an AppExpert application in the Applications node in the configuration utility, a corresponding Access Gateway application is automatically created in the Access Gateway Applications node. Additionally, a rule that uses the AppExpert application's configured public endpoint is automatically created for the Access Gateway application entry. If multiple endpoints are configured for the AppExpert application, the rule includes all the configured public endpoints. The NetScaler appliance uses this rule to apply any configured Access Gateway policies to the traffic received at the AppExpert application's public endpoint. Traffic received at the AppExpert application's public endpoint is first evaluated against the Access Gateway policies and then evaluated against the policies configured for AppExpert application's application units.

The rule that is automatically created for the Authorization, Traffic, and TCP Compression policies for an Access Gateway application is a classic expression that uses the public endpoint that is configured for the AppExpert application. The rule that is created for the Clientless Access policies for an Access Gateway application is an advanced expression that also uses the public endpoint that is configured for the AppExpert application. Therefore, before you configure Access Gateway policies for an AppExpert application, you must configure public endpoints for the AppExpert application.

When you include the Access Gateway configuration in an application template, deployment-specific information, such as IP address and port information, and the rule that is created from this information are not included in the template.

How a NetScaler Configuration for a File Share Works

On the NetScaler appliance, you can configure Authorization policies for a file share that is hosted on your organization's network.

When you create a file share, you specify a name for the file share and the network path to the file share. In the network path, you can specify either the name of the server or the server IP address. A rule that uses the components of the file share path is automatically created for the file share. This rule enables the appliance to identify requests for files hosted on the file share server. Any Authorization policies that are configured for the file share are applied to incoming requests.

The NetScaler configuration for a file share cannot be saved in AppExpert application templates.

How a NetScaler Configuration for an Intranet Subnet Works

For the intranet subnets that form a part of your network, you can configure policies for Authorization, Traffic, and TCP Compression on the NetScaler appliance. When adding an intranet subnet, you specify the IP address and the netmask of the intranet subnet. A rule that uses these two parameters is automatically created for the intranet subnet. The appliance applies the configured policies to any request that has a destination IP address and netmask set to the subnet's IP address and netmask, respectively.

The NetScaler configuration for an intranet subnet cannot be saved in AppExpert application templates.

How the Other Resources Category Works

The Other Resources category enables you to configure Access Gateway policies for any network resource by using a rule of your choice. When you configure the NetScaler appliance to process requests for the network resource, you configure a classic expression to identify the requests that are associated with the network resource. You can configure Authorization, Traffic, Clientless Access, and TCP Compression policies for a network resource in Other Resources. The NetScaler appliance applies the configured Access Gateway policies to any requests that match the configured rule.

The NetScaler configuration for a network resource in Other Resources cannot be saved in AppExpert application templates.

Entity Naming Conventions

The Access Gateway Applications feature enforces a naming convention for some of the entities that you create in this feature. For example, the names of the profiles that you create for Traffic policies for an intranet subnet always begin with a string that consists of the name of the intranet subnet followed by an underscore (_). The name that you provide for the entity is appended to this string. If the name of a subnet is "subnet1," the name of the profile begins with "subnet1_." When such a naming convention is required (in the text box in which you type the name of an entity, for example), the user interface automatically inserts the string with which the name of the entity must begin and does not allow you to modify it.

Adding File Shares

When creating a file share, you provide the network path to the file share. Any policies that you configure for a file share use the rule that is automatically created when you created the file share.

To configure a file share

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, click **File Shares**, and then do one of the following:
   - To add a file share, click **File Shares**, and then click **Add**.
   - To modify a file share, click **File Shares**, and then click **Open**.
3. In the **Create File Share** or **Configure File Share** dialog box, do the following:
   a. In the **Name** box, type a name for the file share you are adding. This parameter cannot be changed for an existing file share.
   b. In the **Path** box, type the path to the file share.

      The path to the file share may use either the name of the server or the IP address of the server.
   c. In **Bookmark**, in the **Text to Display** box, type a name for the file share as you would want it to appear on the Access Gateway home page.
   d. Click **Create** or **OK**, and then click **Close**.

Adding Intranet Subnets

You can specify authorization and Traffic policies for traffic that is bound for the intranet subnets that are configured in your network. The rules for these policies are automatically created by using the parameters you specify for the subnet.

To configure an intranet subnet

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, do one of the following:
   - To add an intranet subnet, click **Intranet Subnets**, and then click **Add**.
   - To modify an intranet subnet, click an intranet subnet, and then click **Open**.
3. In the **Create Intranet Subnet** or **Configure Intranet Subnet** dialog box, do the following:
   a. In the **Name** box, type a name for the intranet subnet you are adding. This parameter cannot be changed for an existing intranet subnet.
   b. In the **IP Address** box, type the IP address of the intranet subnet.
   c. In the **Netmask** box, type the netmask that will be used for the intranet subnet.
   d. Click **Create** or **OK**, and then click **Close**.

Adding Other Resources

For a network resource that you add to Other Resources, you must configure a classic expression that identifies the subset of traffic associated with the resource. For more information about configuring a classic expression, see *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX123868.

To configure a resource in Other Resources

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, do one of the following:
   - To add a resource, click **Other Resources**, and then click **Add**.
   - To modify a resource, click a resource, and then click **Open**.
3. In the **Create Resource** or **Configure Resource** dialog box, do the following:
   a. In the **Name** box, type a name for the resource you are adding. This parameter cannot be changed for an existing resource.
   b. In the **Rule** box, type the rule that will identify the subset of traffic that is associated with the resource you are adding.

      Alternatively, click **Configure**, and then create the rule in the **Create Expression** dialog box.
   c. Click **Create** or **OK**, and then click **Close**.

Configuring Authorization Policies

You can configure Access Gateway authorization policies for AAA users and groups to access a resource.

To configure permissions for a AAA user or group to access a resource

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, in the **Authorization** column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure authorization policies for AAA users and groups.
3. Do one of the following:
   - If the AAA user or group for which you want to configure permissions is already in the **Groups/Users** tree, drag the user or group from the **Groups/Users** tree to the **Users or Groups** node in the <application name> tree. Then, right-click the user or group and click **Allow**.

     If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the <application name> tree, right-click **Users or Groups**, and then click **Add**. In the **Create AAA Group** or **Create AAA User** dialog box, fill in the values, click **Create**, and then click **Close**.

     The user or group is created with the permission set to **Allow**. To change the permission setting, right-click the group or user, and then click the permission setting.
4. Click **Close**.

Configuring Traffic Policies

The traffic policies that you configure for the resources in the Access Gateway Applications node control client connections to the application. You do not have to configure a rule for the resource. The rule created automatically when you create the resource. You only need to associate a request profile with the traffic policy. In the traffic profile, you specify parameters such as the protocol, application time-out, and file type association.

To configure traffic policies for a resource

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, in the **Traffic** column, click the icon provided for the application, file share, intranet subnet, or resource for which you want to configure traffic policies.
3. In the **Configure Traffic Policies** dialog box, do the following:
   - To specify an existing traffic policy, click **Insert Policy**, and then, in the **Policy Name** column, click the name of the policy.

     To configure a new policy, click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**. In the **Create Traffic Policy** dialog box, in the **Name** box, after the underscore (_), type a name for the policy. Then, in **Request Profile**, either select an existing request profile or click **New** to configure a new request profile. You can also select an existing profile and then click **Modify** to modify the profile.

     For more information about configuring a traffic policy or profile, see Access Gateway 9.3, Enterprise Edition at http://edocs.citrix.com/.
   - To modify a policy that you have inserted, in the **Policy Name** column, click the policy name, and then click **Modify Policy**. To modify only the associated profile, in the **Profile** column, click the name of the profile, and then click **Modify Profile**.
   - To regenerate the priorities assigned to the policies, click **Regenerate Priorities**.
   - To specify a new priority value for a policy, in the **Priority** column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes**, and then click **Close**.

Configuring Clientless Access Policies

Clientless access, when configured for a resource on the NetScaler appliance, allows end-users to access the resource without using the Access Gateway client software. Users can use web browsers to access resources such as Outlook Web Access. You configure clientless access for a resource by configuring a clientless access policy that is associated with a clientless access profile.

To configure a clientless access policy for a resource in the Access Gateway Applications node

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.

2. In the details pane, in the **Clientless Access** column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a clientless access policy.

3. In the **Configure Clientless Access Policies** dialog box, do the following:

   - To specify an existing clientless access policy, click **Insert Policy**, and then, in the **Policy Name** column, click the name of the policy.

     To configure a new clientless access policy, click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**. In the **Create Clientless Access Policy** dialog box, in the **Name** box, after the underscore (_), type a name for the policy. Then, in Profile, either select an existing profile or click **New** to configure a new profile. You can also select an existing profile and then click **Modify** to modify the profile.

     For more information about configuring a clientless access policy or profile, see Access Gateway 9.3, Enterprise Edition at http://edocs.citrix.com/.

   - To modify a policy that you have inserted, in the **Policy Name** column, click the policy name, and then click **Modify Policy**. To modify only the associated profile, in the **Profile** column, click the name of the profile, and then click **Modify Profile**.

   - To specify a new priority value for a policy, in the **Priority** column, double-click the assigned priority, and then enter the value you want.

   - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes**, and then click **Close**.

Configuring TCP Compression Policies

You can configure TCP compression policies for an application to increase the performance of the application. TCP compression reduces network latency, reduces bandwidth requirements, and increases the speed of transmission. When configuring a TCP compression policy, you associate a compression action with the policy. The compression action specifies either Compress, GZIP, Deflate, or NoCompress as the compression type. For more information about the compression policies, and compression actions, see Access Gateway 9.3, Enterprise Edition at http://edocs.citrix.com/.

To configure a TCP compression policy for a resource in the Access Gateway Applications node

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.

2. In the details pane, in the **TCP Compression** column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a TCP compression policy.

3. In the **Configure TCP Compression Policies** dialog box, do the following:

   - To specify an existing TCP compression policy, click **Insert Policy**, and then, in the **Policy Name** column, click the name of the policy.

     To create a new TCP compression policy, click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**. In the **Create TCP Compression Policy** dialog box, in the **Policy Name** box, after the underscore ("_"), type a name for the policy. Then, in **Action**, either select an existing action or click **New** and configure a new action. You can also click **View** to view the configured compression type.

     For more information about configuring a TCP compression policy or action, see Access Gateway 9.3, Enterprise Edition at http://edocs.citrix.com/.

   - To modify a policy that you have inserted, in the **Policy Name** column, click the policy name, and then click **Modify Policy**.

   - To regenerate the priorities assigned to the policies, click **Regenerate Priorities**.

   - To specify a new priority value for a policy, in the **Priority** column, double-click the assigned priority, and then enter the value you want.

   - To unbind a policy, click the policy, and then click **Unbind Policy**.

4. Click **Apply Changes**, and then click **Close**.

Configuring Bookmarks

You can configure bookmarks for an application or for a resource that you configure in the Other Resources category if you want the application or resource to be accessible from the Access Gateway home page.

To configure a bookmark for an Access Gateway application or a resource in the Other Resources category

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, click the application or resource for which you want to configure a bookmark, and then click **Configure Bookmark**.
3. In the **Create Bookmark** dialog box, configure values for the parameters.

   For more information about the parameters in the **Create Bookmark** dialog box, see Access Gateway 9.3, Enterprise Edition at http://edocs.citrix.com/.
4. Click **Create**, and then click **Close**.

Entity Templates

An entity template is a collection of configuration information for an individual entity on a Citrix® NetScaler® appliance. It provides a specification and a set of defaults for a configurable NetScaler entity, such as a policy, virtual server, service, or action. By using a template that defines a set of defaults, you can quickly configure multiple entities that require a similar configuration while eliminating several configuration steps.

Entity templates are available only in the configuration utility. You use the NetScaler configuration utility to create, manage, and use any type of entity template. You can share entity templates with other administrators and manage local folders that contain the templates. You can also import entity templates from and export entity templates to your local computer.

Before creating a template, you should be familiar with the configuration of the entity.

Note: You use entity templates to configure individual entities. To configure multiple entities related to a particular Web application, you must use an application template. For more information, see "AppExpert Applications and Templates".

How Entity Templates Work

When you create a template for a NetScaler entity, you specify default values for the entity. You specify what values must be read-only, what values must not be displayed, and what values users can configure. You also configure the pages that compose the template import wizard.

When a user imports the entity template to a NetScaler appliance, a wizard guides the user through the various pages that you configured for the template. The wizard displays the read-only parameter values and prompts the user to specify values for the configurable parameters. After the user follows the instructions in the wizard, the appliance creates the entity with the configured values.

For example, you can create an entity template for HTTP services that provides a text box for a service name and assigns preset values for the service protocol, timeouts, thresholds, and monitors. Later, when you use the template to create new HTTP services, a wizard prompts you for a service name and supplies the preset values that you would otherwise have configured manually.

The procedure for creating entity templates for load balancing virtual servers is different than the AppExpert procedure for creating other entity templates. For more information, see "Creating an Entity Template".

In addition, the procedure for using the template to create the load balancing virtual server entity is different. For more information, see "Creating an Entity from a Template".

Configuring an Entity Template

You can create or modify an entity template either from the AppExpert feature node in the NetScaler configuration utility or from the associated NetScaler feature node for the entity. For example, you can create a content switching virtual server entity template in either the AppExpert feature node or the content switching feature node in the configuration utility.

If you create a template that is not based on an existing entity, you can specify the following options and settings for the template:

- The default value of a parameter.
- Whether the default values are visible to users.
- Whether the default values can be changed by users.
- The number of pages in the entity import wizard, including the page names, text, and available parameters.

  The entities that must be bound to the entity for which the template is being created.

  For example, when you are creating a cache redirection virtual server template, you can specify the policies that you want to bind to the cache redirection virtual servers that you create from the template. However, only binding information is included in the template. The bound entities are not included. If the entity template is imported to another NetScaler appliance, the bound entities must exist on the appliance at import time for the binding to succeed. If none of the bound entities exist on the target appliance, the entity (for which the template was configured) is created without any bindings. If only a subset of the bound entities exist on the target appliance, they are bound to the entity that is created from the template.

When you create a template based on an existing entity, the configuration settings of the entity appear in the template. All bound entities are selected by default, but you can modify bindings as necessary. As in the case of a template that is not based on an existing entity, only binding information is included and not the entities. You can either save the template with the existing configuration settings or use the settings as a basis for creating a new configuration for a template.

However, when you create a template based on an existing load balancing virtual server, all bound entities (services, service groups, policies, actions, and other associated entities) are included in the template. If the bound entities that are included in the template are already configured on the NetScaler appliance to which the template is imported, duplicates are created with names that are generated automatically in a particular format. The duplicate entities are based on the parameter information stored in the entity template.

Creating an Entity Template

The NetScaler AppExpert feature node offers a single location in the configuration utility for creating templates for all types of entities except load balancing virtual servers. Entity templates can also be created in the NetScaler feature node that corresponds to the entity. For example, you can create a content switching virtual server template in the content switching feature node in the configuration utility. Load balancing virtual server templates can be created only from within the NetScaler load balancing feature node. Additionally, you can create a load balancing virtual server template only from an existing load balancing virtual server.

To create an entity template by using the AppExpert feature node

1. In the navigation pane of the NetScaler configuration utility, click **AppExpert**, and then click **Templates**.

2. In the details pane, on the **Entity Templates** tab, do one of the following:

   - To create a new template, click **Add**. In the **Select the Template Type** dialog box, select the template type, and then click **OK**.

   - To create a duplicate of an existing entity template, in the details pane, select the entity template, and then click **Add**.

3. In the **Create...Template** dialog box, follow the instructions to create a template.

   If you are creating a duplicate of an existing entity template, in the **Create...Template** dialog box, on the **Specify Template Name** page, you must change the name of the entity template.

4. Click **Finish**, and then click **Exit**.

To create a load balancing virtual server template

1. In the navigation pane of the NetScaler configuration utility, click **Load balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server, and then click **Create Template**.
3. In the **Create Template** dialog box, provide the following information:
   - **Name.** The name of the template.
   - **Folder.** The location of the template on the appliance. Click **Browse** if you want to change the location.
   - **Configure Variables.** Configure variables for the load balancing template. For more information, see "Configuring Variables in Load Balancing Virtual Server Templates".
   - **Introduction Description.** A description of the configuration of the entity, such as a load balancing virtual server, for which you are creating a template.
   - **Summary Description.** A summary of the configuration or additional instructions for other administrators, such as a description of any additonal steps that need to be followed after the entity is successfully created.
   - **Author.** The creator of the template.
   - **Major.** Major template version.
   - **Minor.** Minor template version.
4. Click **OK**.

To create an entity template by using its corresponding feature node

1. In the navigation pane of the NetScaler configuration utility, select the feature (for example, Content Switching), and then select the entity (for example, Virtual Servers), for which you want to create the entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Create Template**.
3. In the **Create...Template** dialog box, follow the instructions to create a template.
4. Click **Finish**, and then click **Exit**.

Configuring Variables in Load Balancing Virtual Server Templates

Load balancing virtual server templates support the declaration of variables in the configured load balancing parameters and in bound policies and actions. The ability to declare variables enables you to replace preconfigured values with values that suit the environment into which you are importing the template. The Entity Template Wizard, which appears when you import a template, includes a **Specify Variable Values** page on which you can specify appropriate values for the variables that are configured for the entity template. This wizard page appears only when you import a template that is configured with existing variables.

As an example, consider the following expression configured for a policy that is bound to a load balancing virtual server for which you are creating a template. The expression evaluates the value of the Accept-Language header in an HTTP request.

HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")

If you want the value of the header to be configurable at import time, you can specify the string "en-us" as a variable. When importing the template, you can specify a new value for the variable on the **Specify Variable Values** page.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

To configure variables in a load balancing virtual server template

1. In the navigation pane of the NetScaler configuration utility, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, right-click the virtual server that you want to export to a template file, and then click **Create Template**.
3. In the **Create Template** dialog box, modify the default template file name if required, specify the location where you want to save the template, and then click **Configure Variables**.
4. In the **Configure Variables** dialog box, click the tab that lists the entity for which you want to configure a variable, select the entity, and then click **Configure Variables**.
5.  In the **Variables for <Entity Type>: <Entity Name>** dialog box, click the button next to the parameter value or expression in which you want to create a variable.
6. In the **Variables for <Field Name>** dialog box, do the following:

   - To create a variable, in the text box that displays the configured expression or value, select the string that you want to be configurable at import time, and then click **Add**. In the **Create Variable** dialog box, specify a name and a description for the variable, and then click **Create**.

     The name of the variable, its value, and the description you provided appear in the **Available Variables** listing in the dialog box. The name you provide will be the name of the associated field in the template import wizard, and the description will appear as alt text when the user positions the mouse pointer over the field.

   - To modify a variable, in the **Available Variables** list, click the variable, and then click **Open**. In the **Create Variable** dialog box, modify the value and the description, and then click **OK**.

     The new value that you specify will not replace the text selected in the text box that displays the configured expression or value. However, when you import the template, the new value will be displayed as the default value for the variable in the template import wizard.

   - To view all the strings that are assigned to a given variable, in the **Available Variables** listing, click the name of the variable. The strings that are assigned to the variable are highlighted.
   - To view a list of all the parameters, expressions, and actions in which the variable is used, in the **Available Variables** listing, click the variable whose references you want to view, and then click **Show References**.
   - To assign a string to an existing variable, in the text box that displays the expression you configured, select the string you want to assign to an existing variable, right-click the selection, click **Use existing Variable**, and then click the

name of the variable to which you want to assign the string.

If a variable has multiple strings assigned to it, when you specify a new value for the variable during import, all strings assigned to the variable are replaced with the new value.

7. Click **Close**.

Modifying an Entity Template

You can modify only the parameters, bindings, and pages configured for a template. The name and location of the template specified when the template was created cannot be changed. In load balancing virtual server templates, since the load balancing virtual server configuration is saved in the load balancing virtual server template, there is no option to modify the load balancing virtual server template.

To modify an entity template by using the AppExpert feature node

1. In the navigation pane, expand **AppExpert**, and then click **Templates**.
2. In the details pane, on the **Entity Templates** tab, select the template you want to change, and then click **Open**.
3. In the **Modify...Template** dialog box, follow the instructions to modify a template.
4. Click **Finish**, and then click **Exit**.

To modify an entity template by using its corresponding feature node

1. In the navigation pane, select the feature (for example, Content Switching), and then select the entity (for example, Virtual Servers) for which you want to modify the entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the **Manage <feature entity name> Entity Templates** dialog box, select the template that you want to modify, and then click **Modify**.
4. In the **Modify <template name> Template** dialog box, follow the instructions to modify a template.
5. Click **Finish**, and then click **Exit**.
6. Click **Close**.

Deleting an Entity Template

Deleting an entity template does not affect any objects that have been created by using the template. You can delete a load balancing virtual server template only from the AppExpert feature node.

To delete an entity template by using the AppExpert feature node

1. In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **Templates**.
2. In the details pane, on the **Entity Templates** tab, click the template you want to delete, and then click **Remove**.

To delete an entity template by using its corresponding feature node

1. In the navigation pane of the NetScaler configuration utility, select the feature (for example, Content Switching) and then select the entity (for example, Virtual Servers), for which you want to delete the entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the **Manage...Entity Templates** dialog box, select the template that you want to delete, and then click **Delete**.

Creating an Entity from a Template

You can create an entity from an entity template either from the AppExpert feature node in the NetScaler configuration utility or from the NetScaler feature node that corresponds to the type of entity that you want to create. For example, you can create a content switching virtual server from a template with either the AppExpert feature node or the content switching feature node in the configuration utility.

The procedure for creating a load balancing virtual server from a template is different than the AppExpert procedure for creating other entities from templates.

After you create an instance of an entity using an entity template, you can configure it in the same way that you would any other object of that type, such as by using the configuration utility or the command line.

To create an entity from a template by using the AppExpert feature node

1. In the navigation pane, expand **AppExpert**, and then click **Templates**.
2. In the details pane, on the **Entity Templates** tab, click the template that you want to use, and then click **Use Template**.
3. In the <Entity Template Name> wizard, follow the instructions to create the entity on the NetScaler.
4. Click **Finish**, and then click **Exit**.

To create an entity from a template by using its corresponding feature node

1. In the navigation pane, expand a feature node (for example, **Content Switching**), and then click an entity subnode (for example, **Virtual Servers**).
2. At the top of the details pane, click **Entity Templates**, and then click **Use Template**.
3. Click the name of the template that you want to use.
4. In the Use <template name> Template wizard, follow the instructions to create the entity.

   Only templates that match the current context are displayed. For example, in the details pane for content switching virtual servers, only entity templates for content switching virtual servers appear, if configured.
5. Click **Finish**, and then click **Exit**.

To create a load balancing virtual server by using a load balancing virtual server template

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Use Template**.
3. In the Entity Template Wizard, follow the instructions to create a load balancing virtual server on the NetScaler.

   Only templates that match the current context are displayed. For example, when you click **Browse (Appliance)**, only entity templates for load balancing virtual servers appear, if configured.
4. Click **Finish**, and then click **Exit**.

   Note: The Entity Template Wizard includes a Specify Variable Values page on which you can specify new values for variables. For more information about configuring variables in load balancing virtual server templates, see "Configuring Variables in Load Balancing Virtual Server Templates".

Managing Entity Template Folders

You can create, rename, and rearrange local folders for organizing and storing templates for all entities except load balancing virtual servers. Load balancing virtual server templates are stored in the /nsconfig/nstemplates/entities/lb vserver folder.

To organize entity template folders

1. In the navigation pane of the NetScaler configuration utility, expand a feature node (for example, **Content Switching**), and then click a subnode (for example, **Virtual Servers**) that corresponds to the entity template that you want to manage.

2. At the top of the details pane, click **Entity Templates**, and then click **Manage Templates**.

3. In the **Manage...Entity Templates** dialog box, do one of the following:

   - To create a new folder under the selected folder in the folder tree, click **Create Folder**.

     To change the name of a folder, select the folder and click **Rename**.

     You can also click the folder that you want to rename, and then press F2. You cannot rename the top-level default folder.

     To remove the folder, select the folder and click **Delete**.

     You can also click the folder that you want to remove, and then press the Delete key. You cannot remove the top-level default folder.
4. Click **Close.**

Uploading and Downloading Entity Templates

You can import the entity templates that are stored on your local computer. You can also download entity templates from the NetScaler appliance to your local computer and then import them to other NetScaler appliances.

Note: You cannot upload or download load balancing virtual server templates.

To upload an entity template to the NetScaler appliance

1. In the navigation pane of the NetScaler configuration utility, expand a feature node (for example, **Content Switching**), and then click a subnode (for example, **Virtual Servers**) for which you want to upload an entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the **Manage...Entity Templates** dialog box, click the top-level folder, and then click **Upload**.
4. In the **Upload Entity Template** dialog box, navigate to the template file that you want to upload, and then click **Select**.
5. Click **Close**.

To download an entity template from the NetScaler appliance

1. In the navigation pane of the NetScaler configuration utility, expand a feature node (for example, **Content Switching**), and then click a subnode (for example, **Virtual Servers**) for which you want to upload an entity template.

2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.

3. In the **Manage...Entity Templates** dialog box, click the template that you want to download, and then click **Download.**

4. In the **Download Entity Template** dialog box, navigate to the location at which you want to save the template on your local computer, enter a file name, and then click **Save**.

5. Click **Close.**

Citrix Application Firewall

The following topics cover installation and configuration of the Citrix Application Firewall feature.

Introduction	An overview of web application security and how the application firewall works.
Configuration	How to configure the application firewall to protect a web site, a web service, or a web 2.0 site.
Signatures	A detailed description of the signatures feature and how to configure the signatures, add signatures from a supported vulnerability scanning tool, and define your own signatures, with examples.
Advanced Protections	A detailed description of all of the application firewall security checks, with configuration information and examples.
Policies	A description of how policies are used when configuring the application firewall, with examples of useful policies.
Imports	A description of how the application firewall uses different types of imported files, and how to import and export files.
Global Configuration	A description of application firewall features that apply to all profiles, and how to configure them.
Use Cases	Extended examples that demonstrate how to set up the application firewall to best protect specific types of more complex web sites and web services.
Logs, Statistics, and Reports	How to access and use the application firewall logs, the statistics, and the reports to assist in configuring the application firewall.

Introduction

The Citrix® Application Firewall™ prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information. It does so by filtering both requests and responses, examining them for evidence of malicious activity, and blocking those that exhibit such activity. Your site is protected not only from common types of attacks, but also from new, as yet unknown attacks. In addition to protecting web servers and web sites from unauthorized access and misuse by hackers and malicious programs, the application firewall provides protection against security vulnerabilities in legacy CGI code or scripts, other web frameworks, web server software, and the underlying operating systems.

The Citrix Application Firewall is available as a stand-alone appliance, or as a feature on a Citrix NetScaler® appliance or NetScaler VPX™ virtual appliance. In the application firewall documentation, the term *application firewall appliance* refers to the platform on which the application firewall is running, regardless of whether that platform is a dedicated firewall appliance, a NetScaler appliance on which other features have also been configured, or a NetScaler VPX virtual appliance,

To use the application firewall, you must create at least one security configuration to block connections that violate the rules that you set for your protected web sites. The number of security configurations that you might want to create depends on the complexity of your web site. In many cases, one is sufficient. You can probably use the defaults for the global settings, which affect all security configurations, but you can change the global settings if necessary.

Introduction

The Citrix® Application Firewall™ prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information. It does so by filtering both requests and responses, examining them for evidence of malicious activity, and blocking those that exhibit such activity. Your site is protected not only from common types of attacks, but also from new, as yet unknown attacks. In addition to protecting web servers and web sites from unauthorized access and misuse by hackers and malicious programs, the application firewall provides protection against security vulnerabilities in legacy CGI code or scripts, other web frameworks, web server software, and the underlying operating systems.

The Citrix Application Firewall is available as a stand-alone appliance, or as a feature on a Citrix NetScaler® appliance or NetScaler VPX™ virtual appliance. In the application firewall documentation, the term *application firewall appliance* refers to the platform on which the application firewall is running, regardless of whether that platform is a dedicated firewall appliance, a NetScaler appliance on which other features have also been configured, or a NetScaler VPX virtual appliance,

To use the application firewall, you must create at least one security configuration to block connections that violate the rules that you set for your protected web sites. The number of security configurations that you might want to create depends on the complexity of your web site. In many cases, one is sufficient. You can probably use the defaults for the global settings, which affect all security configurations, but you can change the global settings if necessary.

Web Application Security

Web application security is that part of network security that covers computers and programs that communicate by using the HTTP and HTTPS protocols. This is an extremely broad area in which security flaws and weaknesses abound. Operating systems on both servers and clients have security issues and are vulnerable to attack. Web server software and web site enabling technologies such as CGI, Java and JavaScript have underlying vulnerabilities. Browsers and other client applications that communicate with web-enabled applications also have vulnerabilities. Web sites that use any technology but the simplest of HTML, including any site that allows interaction with visitors, often have vulnerabilities of their own.

In the past, a breach in security was often just an annoyance, but today that is seldom the case. For example, attacks in which a hacker gained access to a web server and made unauthorized modifications to (*defaced*) a web site used to be common. They were usually launched by hackers who had no motivation beyond demonstrating their skills to fellow hackers. Most current security breaches, however, are motivated by a desire for money. The majority attempt to accomplish one or both of the following goals: to obtain sensitive and potentially valuable private information, or to obtain unauthorized access to and control of a web site or web server.

Certain forms of web attacks focus on obtaining private information. These attacks are often possible even against web sites that are secure enough to prevent an attacker from taking full control. The information that an attacker can obtain from a web site can include customer names, addresses, phone numbers, social security numbers, credit card numbers, medical records, and other private information. The attacker can then use this information or sell it to others. Much of the information obtained by such attacks is protected by law, and all of it by custom and expectation. A breach of this type can have extremely serious consequences for customers whose private information is compromised. At best, these customers will have to exercise vigilance to prevent others from abusing their credit cards, opening unauthorized credit accounts in their name, or appropriating their identities outright (identity theft). At worst, the customers may face ruined credit ratings or even be blamed for criminal activities in which they had no part.

Other web attacks are aimed at obtaining control of (or *compromising*) your web site or the server on which it operates, or both. A hacker who gains control of a web site or server can use it to host unauthorized content, act as a proxy for content hosted on another web server, provide SMTP services to send unsolicited bulk email, or provide DNS services to support such activities on other compromised web servers. Most web sites that are hosted on compromised web servers promote questionable or outright fraudulent businesses. For example, the majority of phishing web sites and child pornography web sites are hosted on compromised web servers.

Protecting your web sites and web services against these attacks requires a multilayered defense capable of both blocking known attacks with identifiable characteristics and protecting against unknown attacks, which can often be detected because they look different from the normal traffic to your web sites and web services.

Known Web Attacks

The first line of defense for your web sites is protection against the large number of attacks that are known to exist and have been observed and analyzed by web security experts. Common types of attacks against HTML-based web sites include:

- **Buffer overflow attacks.** Sending an extremely long URL, extremely long cookie, or other extremely long bit of information to a web server in hopes of causing it or the underlying operating system to hang, crash, or provide the attacker with access to the underlying operating system. A buffer overflow attack can be used to gain access to unauthorized information, to compromise a web server, or both.

- **Cookie security attacks.** Sending a modified cookie to a web server, usually in hopes of obtaining access to unauthorized content by using falsified credentials.

- **Forceful browsing.** Accessing URLs on a web site directly, without navigating to the URLs by means of hyperlinks on the home page or other common start URLs on the web site. Individual instances of forceful browsing may simply indicate a user who bookmarked a page on your web site, but repeated attempts to access nonexistent content, or content that users should never access directly, often represent an attack on web site security. Forceful browsing is normally used to gain access to unauthorized information, but can also be combined with a buffer overflow attack in an attempt to compromise your server.

- **Web form security attacks.** Sending inappropriate content to your web site in a web form. Inappropriate content can include modified hidden fields, HTML or code in a field intended for alphanumeric data only, an overly long string in a field that accepts only a short string, an alphanumeric string in a field that accepts only an integer, and a wide variety of other data that your web site does not expect to receive in that web form. A web form security attack can be used either to obtain unauthorized information from your web site or to compromise the web site outright, usually when combined with a buffer overflow attack.

Two specialized types of attacks on web form security deserve special mention:

- **SQL injection attacks.** Sending an active SQL command or commands in a web form or as part of a URL, with the goal of causing an SQL database to execute the command or commands. SQL injection attacks are normally used to obtain unauthorized information.

- **Cross-site scripting attacks.** Using a URL or a script on a web page to violate the same-origin policy, which forbids any script from obtaining properties from or modifying any content on a different web site. Since scripts can obtain information and modify files on your web site, allowing a script access to content on a different web site can provide an attacker the means to obtain unauthorized information, to compromise a web server, or both.

Attacks against XML-based web services normally fall into at least one of the following two categories: attempts to send inappropriate content to a web service, or attempts to breach security on a web service. Common types of attacks against XML-based web services include:

- **Malicious code or objects.** XML requests that contain code or objects that can either directly obtain sensitive information or can give an attacker control of the web service or underlying server.

- **Badly-formed XML requests.** XML requests that do not conform to the W3C XML specification, and that can therefore breach security on an insecure web service.

- **Denial of service (DoS) attacks.** XML requests that are sent repeatedly and in high volume, with the intent of overwhelming the targeted web service and denying legitimate users access to it.

In addition to standard XML-based attacks, XML web services and Web 2.0 sites are also vulnerable to SQL injection and cross-site scripting attacks, as described below:

- **SQL injection attacks.** Sending an active SQL command or commands in an XML-based request, with the goal of causing an SQL database to execute that command or commands. As with HTML SQL injection attacks, XML SQL injection attacks are normally used to obtain unauthorized information.

- **Cross-site scripting attacks.** Using a script included in an XML based application to violate the same-origin policy, which does not allow any script to obtain properties from or modify any content on a different application. Since scripts can obtain information and modify files by using your XML application, allowing a script access to content belonging to a different application can give an attacker the means to obtain unauthorized information, to compromise the application, or both.

Known web attacks can usually be stopped by filtering web site traffic for specific characteristics (*signatures*) that always appear for a specific attack and should never appear in legitimate traffic. This approach has the advantages of requiring relatively few resources and posing relatively little risk of false positives. It is therefore a valuable tool in fighting attacks on web sites and web services, and configuring basic signature protections that intercept most known web attacks is easy to do.

Unknown Web Attacks

The greatest threat against web sites and applications does not come from known attacks, but from unknown attacks. Most unknown attacks fall into one of two categories: newly-launched attacks for which security firms have not yet developed an effective defense (*zero day* attacks), and carefully-targeted attacks on a specific web site or web service rather than many web sites or web services (*spear* attacks). These attacks, like known attacks, are usually intended to obtain sensitive private information, compromise the web site or web service and allow it to be used for further attacks, or both of those goals.

Zero-day attacks are a major threat to all users. These attacks are usually of the same types as known attacks; zero-day attacks often involve injected SQL, a cross-site script, a cross-site request forgery, or another type of attack similar to known attacks. In most cases, they target vulnerabilities that the developers of the targeted software, web site, or web service either are unaware of or have just learned about. Security firms have therefore usually not developed defenses against these attacks, and even if they have, users have usually not obtained and installed the patches or performed the workarounds necessary to protect against these attacks. The time between discovery of a zero-day attack and availability of a defense (the *vulnerability window*) is shrinking, but perpetrators can still count on hours or even days in which many web sites and web services lack any specific protection against the attack.

Spear attacks are a major threat, but to a more select group of users. A common type of spear attack, a spear phish, is usually targeted at customers of a specific bank or financial institution, or (less commonly) at employees of a specific company or organization. Unlike other phishes, which are often crudely written forgeries that a user with any familiarity with the actual communications of that bank or financial institution can recognize, spear phishes are letter perfect and extremely convincing. They can contain information specific to the individual that, at first look, no stranger should know or be able to obtain. The spear phisher is therefore able to convince his or her target to provide the requested information, which the phisher can then use to loot accounts, to process illegitimately obtained money from other sources, or to gain access to other, even more sensitive information.

Both of these types of attack have certain characteristics that can usually be detected, although not by using static patterns that look for specific characteristics, as do standard signatures. Detecting these types of attacks requires more sophisticated and more resource-intensive approaches, such as heuristic filtering and positive security model systems. Heuristic filtering looks, not for specific patterns, but for patterns of behaviors. Positive security model systems model the normal behavior of the web site or web service that they are protecting, and then block connections that do not fit within that model of normal use. URL based and web-form based security checks profile normal use of your web sites, and then control how users interact with your web sites, using both heuristics and positive security to block anomalous or unexpected traffic. Both heuristic and positive security, properly designed and deployed, can catch most attacks that signatures miss. However, they require considerably more resources than do signatures, and you must spend some time configuring them properly to avoid false positives. They are therefore usually used, not as the primary line of defense, but as backups to signatures or other less resource-intensive approaches.

By configuring these advanced protections in addition to signatures, you create a hybrid security model, which enables the application firewall to provide comprehensive protection against both known and unknown attacks.

How The Application Firewall Works

When you install the application firewall, you create an initial *security configuration*, which consists of a *policy*, a *profile*, and a *signatures object*. The policy is a rule that identifies the traffic to be filtered, and the profile identifies the patterns and types of behavior to allow or block when the traffic is filtered. The simplest patterns, which are called *signatures*, are not specified within the profile, but in a signatures object that is associated with the profile.

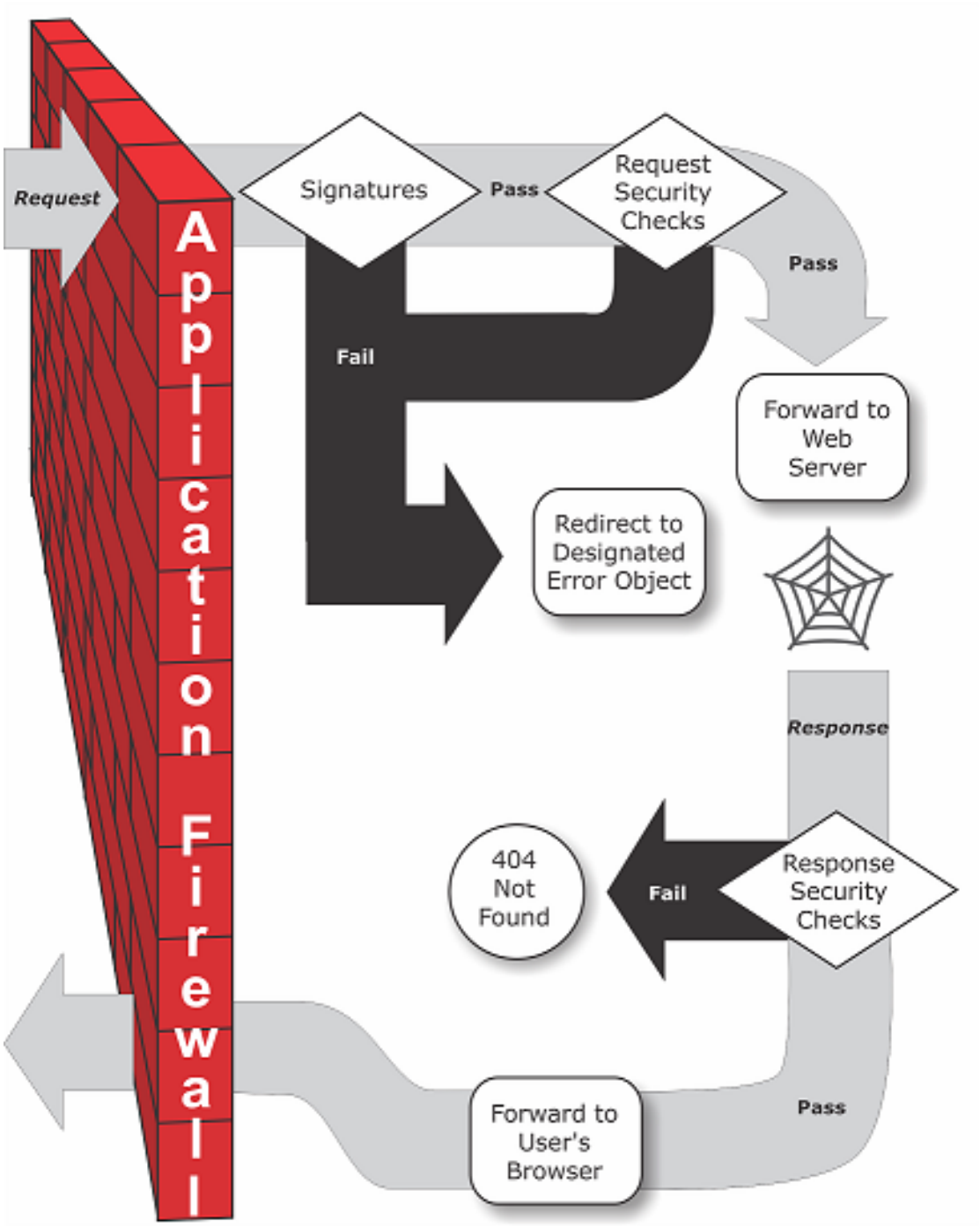
A signature is a string or pattern that matches a known type of attack. The application firewall contains over a thousand signatures in seven categories, each directed at attacks on specific types of web servers and web content. Citrix updates the list with new signatures as new threats are identified. During configuration, you specify the signature categories that are appropriate for the web servers and content that you need to protect. Signatures provide good basic protection with low processing overhead. If your applications have special vulnerabilities or you detect an attack against them for which no signature exists, you can add your own signatures.

The more advanced protections are called *security checks*. A security check is a more rigorous, algorithmic inspection of a request for specific patterns or types of behavior that might indicate an attack or constitute a threat to your protected web sites and web services. It can, for example, identify a request that attempts to perform a certain type of operation that might breach security, or a response that includes sensitive private information such as a social security number or credit card number. During configuration, you specify the security checks that are appropriate for the web servers and content that you need to protect. The security checks are restrictive. Many of them can block legitimate requests and responses if you do not add the appropriate exceptions (*relaxations*) when configuring them. Identifying the needed exceptions is not difficult if you use the adaptive learning feature, which observes normal use of your web site and creates recommended exceptions.

The application firewall can be installed as either a Layer 3 network device or a Layer 2 network bridge between your servers and your users, usually behind your company's router or firewall. It must be installed in a location where it can intercept traffic between the web servers that you want to protect and the hub or switch through which users access those web servers. You then configure the network to send requests to the application firewall instead of directly to your web servers, and responses to the application firewall instead of directly to your users. The application firewall filters that traffic before forwarding it to its final destination, using both its internal rule set and your additions and modifications. It blocks or renders harmless any activity that it detects as harmful, and then forwards the remaining traffic to the web server. The following figure provides an overview of the filtering process.

Note: The figure omits the application of a policy to incoming traffic. It illustrates a security configuration in which the policy is to process all requests. Also, in this configuration, a signatures object has been configured and associated with the profile, and security checks have been configured in the profile.

Figure 1. A Flowchart of Application Firewall Filtering



As the figure shows, when a user requests a URL on a protected web site, the application firewall first examines the request to ensure that it does not match a signature. If the request matches a signature, the application firewall either displays the *error object* (a web page that is located on the application firewall appliance and which you can configure by using the imports feature) or forwards the request to the designated error URL (the *error page*). Signatures do not require as many resources as do security checks, so detecting and stopping attacks that are detected by a signature before running any of the security checks reduces the load on the server.

If a request passes signature inspection, the application firewall applies the request security checks that have been enabled. The request security checks verify that the request is appropriate for your web site or web service and does not contain material that might

pose a threat. For example, security checks examine the request for signs indicating that it might be of an unexpected type, request unexpected content, or contain unexpected and possibly malicious web form data, SQL commands, or scripts. If the request fails a security check, the application firewall either sanitizes the request and then sends it back to the NetScaler appliance (or NetScaler VPX virtual appliance), or displays the error object. If the request passes the security checks, it is sent back to the NetScaler appliance, which completes any other processing and forwards the request to the protected web server.

When the web site or web service sends a response to the user, the application firewall applies the response security checks that have been enabled. The response security checks examine the response for leaks of sensitive private information, signs of web site defacement, or other content that should not be present. If the response fails a security check, the application firewall either removes the content that should not be present or blocks the response. If the response passes the security checks, it is sent back to the NetScaler appliance, which forwards it to the user.

Application Firewall Features

The basic application firewall features are policies, profiles, and signatures, which provide a hybrid security model as described in Known Web Attacks, Unknown Web Attacks, and How the Application Firewall Works. Of special note is the learning feature, which observes traffic to your protected applications and recommends appropriate configuration settings for certain security checks.

The imports feature manages files that you upload to the application firewall. These files are then used by the application firewall in various security checks, or when responding to a connection that matches a security check.

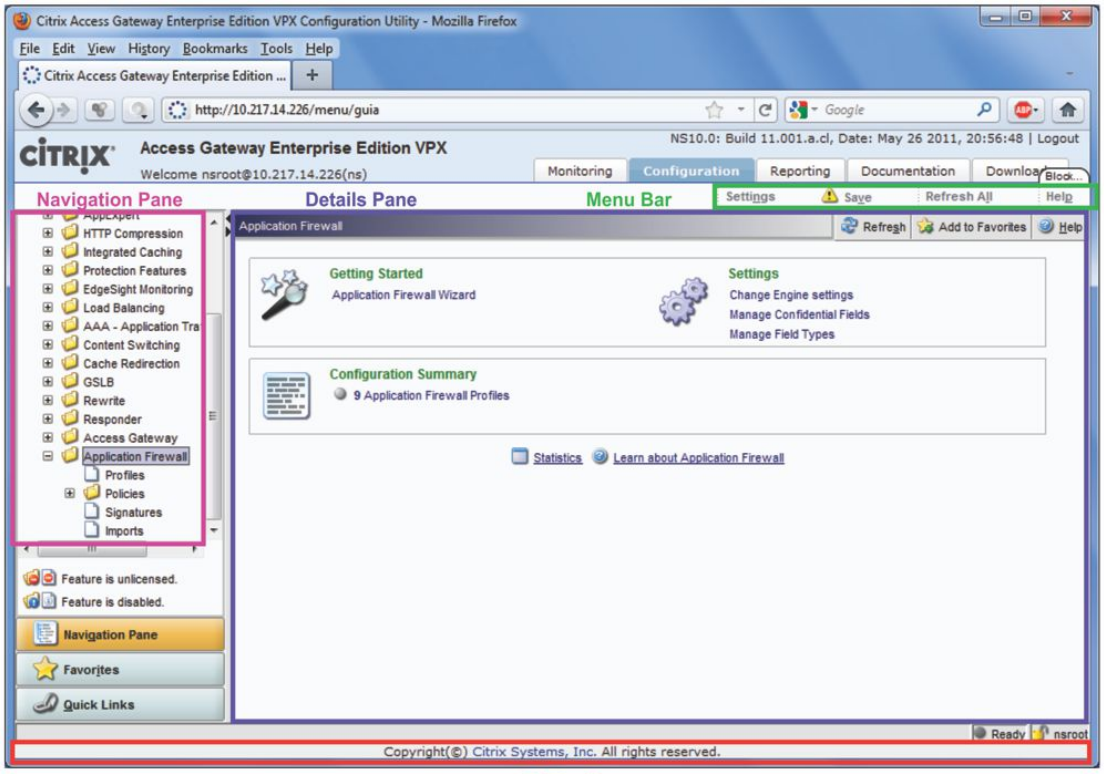
You can use the logs, statistics, and reports features to evaluate the performance of the application firewall and identify possible needs for additional protections.

The Application Firewall User Interfaces

All models in the Citrix NetScaler Application Delivery product line can be configured and managed from the Citrix NetScaler command line interface or the web-based configuration utility. However, the configuration utility provides a more complete interface. Not all application firewall configuration tasks can be performed at the command line. Also, the configuration utility provides access to a wizard that reduces the complexity of configuring the applicaton firewall. Unlike most wizards, the application firewall wizard can serve as your primay interface to the application firewall.

The command line interface is a modified UNIX shell based on the FreeBSD `bash` shell. To configure the Application Firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. For instructions for using the command line inerface, see the *Citrix NetScaler Command Reference Guide*

Figure 1 shows the configuration utility's System Overview screen.



The screen has two main areas. The panel on the left, called the *navigation pane*, contains a navigation tree, with which you navigate to the screens on which you configure the features that are installed on your appliance. The screens to which you navigate appear to the right of the navigation pane, in the *details pane*.

When you access the configuration utility, the details pane displays the System Overview screen, as shown in Figure 1. If, in the navigation pane, you click plus sign next to the Application Firewall folder, the **Application Firewall** node *expands* to include the main

application firewall elements that you can configure. If you click the first element, **Profiles**, the details pane displays the configured profiles, if any profiles have been configured. At the bottom of the details pane, you can click **Add** to configure a new profile. Other buttons at the bottom of the details pane are grayed out until you select an existing profile. Screens for the other elements work in the same way.

If, instead of expanding the application firewall node, you click the node itself, the details pane displays different options, one of which is the application firewall wizard. Citrix recommends that you use the wizard for initial configuration, and many users use it almost exclusively. It includes most of the functionality that is available elsewhere in the configuration utility.

For information and instructions on accessing the configuration utility, see the *Citrix NetScaler Getting Started Guide*.

Configuring the Application Firewall

To configure the Citrix Application Firewall™, Citrix recommends that you use a browser to connect to the configuration utility. When the connection is established, verify that the application firewall is enabled, and then run the application firewall wizard, which prompts you for configuration information. You do not have to provide all of the requested information the first time you use the wizard. Instead, you can accept default settings, perform a few relatively straightforward configuration tasks to enable important features, and then allow the application firewall to collect important information to help you complete the configuration.

For example, when the wizard prompts you to specify a rule for selecting the traffic to be processed, you can accept the default, which selects all traffic. When it presents you with a list of signatures, you can enable the appropriate categories of signatures and turn on the collection of statistics for those signatures. For this initial configuration, you can skip the advanced protections (*security checks*). The wizard automatically creates the appropriate policy, signatures object, and profile (collectively, the *security configuration*) , and binds the policy to global. The application firewall then begins filtering connections to your protected web sites, logging any connections that match one or more of the signatures that you enabled, and collecting statistics about the connections that each signature matches. After the application firewall processes some traffic, you can run the wizard again and examine the logs and statistics to see if any of the signatures that you have enabled are matching legitimate traffic. After determining which signatures are identifying the traffic that you want to block, you can enable blocking for those signatures. If your web site or web service is not complex, does not use SQL, and does not have access to sensitive private information, this basic security configuration will probably provide adequate protection.

You may need additional protection if, for example, your web site is dynamic. Content that uses scripts may need protection against cross-site scripting attacks. Web content that uses SQL—such as shopping carts, many blogs, and most content management systems—may need protection against SQL injection attacks. Web sites and web services that collect sensitive private information such as social security numbers or credit card numbers may require protection against unintentional exposure of that information. Certain types of web-server or XML-server software may require protection from types of attacks tailored to that software. Another consideration is that specific elements of your web sites or web services may require different protection than do other elements. Examining the application firewall logs and statistics can help you identify the additional protections that you might need.

After deciding which advanced protections are needed for your web sites and web services, you can run the wizard again to configure those protections. Certain security checks require that you enter exceptions (*relaxations*) to prevent the check from blocking legitimate traffic. You can do so manually, but it is usually easier to enable the adaptive learning feature and allow it to recommend the necessary relaxations. You can use the wizard as many times as necessary to enhance your basic security configuration and/or create additional security configurations.

The wizard automates some tasks that you would have to perform manually if you did not use the wizard. It automatically creates a policy, a signatures object, and a profile, and assigns them the name that you provided when you were prompted for the name of your configuration. The wizard also adds your advanced-protection settings to the profile, binds

the signatures object to the profile, associates the profile with the policy, and puts the policy into effect by binding it to Global.

A few tasks cannot be performed in the wizard. You cannot use the wizard to bind a profile to a bind point other than Global. If you want the profile to apply to only a specific part of your configuration, you must manually configure the binding. You cannot configure the engine settings or certain other global configuration options in the wizard. While you can configure any of the advanced protection settings in the wizard, if you want to modify a specific setting in a single security check, it may be easier to do so on the manual configuration screens in the configuration utility.

Enabling the Application Firewall

Before you can create an application firewall security configuration, you must make sure that the application firewall feature is enabled.

- If you are configuring a dedicated Citrix Application Firewall appliance, the feature is already enabled. You do not have to perform either of the procedures described here.
- If you have a Citrix NetScaler appliance but have not previously configured the application firewall, you need to enable the application firewall feature before you configure it.
- If you are upgrading a NetScaler appliance from a previous version of the NetScaler operating system to the current version, you may need to enable the application firewall feature before you configure it.
- If you are installing a new application firewall appliance or NetScaler appliance, you do not need to perform this procedure.

Note: If you are upgrading a NetScaler appliance or NetScaler VPX virtual appliance from a previous version, you may need to update the licenses on your appliance before you can enable this feature.

You can enable the application firewall by using the NetScaler command line or the configuration utility.

To enable the application firewall by using the NetScaler command line

At the NetScaler command prompt, type:

enable ns feature **AppFW**

To enable the application firewall by using the configuration utility

1. In the navigation pane, expand **System** and click **Settings**.
2. In the **Settings** pane, under Modes & Features, click **basic features**.
3. In the **Configure Basic Features** dialog box, select the **Application Firewall** check box.
4. Click **OK**.

The Application Firewall Wizard

Unlike most wizards, the Application Firewall wizard is designed not just to simplify the initial configuration process, but also to modify previously created configurations and to maintain your Application Firewall setup. A typical user runs the wizard multiple times, skipping some of the screens each time.

Opening the Wizard

To run the Application Firewall wizard, first open the configuration utility. Next, in the navigation pane, expand Application Firewall, and then in the details pane click Application Firewall Wizard. (For more information about the configuration utility, see The NetScaler Configuration Utility.) Then:

1. In the navigation pane, click the **Application Firewall** folder icon.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**. The first screen of the wizard appears.
3. To advance to the next screen, click **Next**.

The Wizard Screens

The Application Firewall wizard displays the following screens, in the following order:

1. **Introduction screen.** Provides an introduction to the Application Firewall wizard. There is nothing that you can configure on this screen.
2. **Specify Name screen.** On this screen, when creating a new security configuration, you specify the name that the wizard is to assign to the configuration. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. Choose a name that makes it easy for others to tell what content your new security configuration protects.

   Note: Because the wizard uses this name for both the policy and the profile, it is limited to 31 characters. Manually created policies can have names up to 127 characters in length.

   When creating an existing configuration, you select **Modify Existing Configuration** and then, in the **Name** drop-down list, select the name of the existing configuration that you want to modify.

Note: Only policies that are bound to global or to a bind point appear in this list; you cannot modify an unbound policy by using the Application Firewall wizard. You must either manually bind it to Global or a bind point, or modify it manually. (For manual modification, in the configuration utility's **Application Firewall --> Policies --> Firewall** pane, select the policy and click **Open**).

You also select a profile type on this screen. The profile type determines the types of advanced protection (*security checks*) that can be configured. Because certain kinds of content are not vulnerable to certain types of security threats, restricting the list of available checks saves time during configuration. The types of Application Firewall profiles are:

- **Web Application (HTML).** Any HTML-based Web site that does not use XML or Web 2.0 technologies.
- **XML Application (XML, SOAP).** Any XML-based Web service.
- **Web 2.0 Application (HTML, XML, REST).** Any Web 2.0 site that combines HTML and XML-based content, such as an ATOM-based site, a blog, an RSS feed, or a wiki.

Note: If you are unsure which type of content is used on your Web site, you can choose Web 2.0 Application to ensure that you protect all types of Web application content.

3. **Specify Rule screen.** On this screen, you specify the policy rule that defines the traffic to be examined by this security configuration. If you are creating an initial configuration to protect your Web sites and Web services, you can simply accept the default value, `true`, which selects all web traffic .

   If you want this security configuration to examine, not all HTTP traffic that is routed through the appliance, but specific traffic, you can write a policy rule specifying the traffic that you want it to examine. Rules are written in Citrix NetScaler expressions language, which is a fully functional object-oriented programming language.

   - For a simple description of using the NetScaler expressions syntax to create Application Firewall rules, and a list of useful rules, see Firewall Policies.
   - For a detailed explanation of how to create policy rules in NetScaler expressions syntax, see the *Citrix Policy Configuration and Reference Guide*.

4. **Select Signature Protections screen.** On this screen, you select the categories of signatures that you want to use to protect your web sites and web services. The default categories are:

   - **CGI.** Protection against attacks on web sites that use CGI scripts in any language, including PERL scripts, Unix shell scripts, and Python scripts.
   - **Cold Fusion.** Protection against attacks on web sites that use the Adobe Systems® ColdFusion® Web development platform.
   -

FrontPage. Protection against attacks on web sites that use the Microsoft® FrontPage® Web development platform.

- **PHP.** Protection against attacks on web sites that use the PHP open-source Web development scripting language.
- **Client side.** Protection against attacks on client-side tools used to access your protected web sites, such as Microsoft Internet Explorer, Mozilla Firefox, the Opera browser, and the Adobe Acrobat Reader.
- **Microsoft IIS.** Protection against attacks on Web sites that run the Microsoft Internet Information Server (IIS).
- **Miscellaneous.** Protection against attacks on other server-side tools, such as Web servers and database servers.

If you are creating a new security configuration, the signature categories that you select are enabled, and by default they are recorded in a new signatures object. The new signatures object is assigned the same name that you entered on the **Specify name** screen as the name of the security configuration.

If you have previously configured signatures objects and want to use one of them as the signatures object associated with the security configuration that you are creating, click **Select Existing Signature** and select a signatures object from the **Signatures** list.

If you are modifying an existing security configuration, you can click **Select Existing Signature** and assign a different signatures object to the security configuration.

5. **Select Signature Actions screen.**On this screen, you select the actions associated with the signature categories that you selected on the **Select signature protections** screen. If you are creating an initial configuration, you might want to accept the defaults, which enable the Log and Stats actions but not the Block action. You can decide later, after reviewing the collected logs and statistics, which signatures you should use to block traffic, and then enable the Block action for those signatures. Signatures are designed to catch specific known attacks on your web sites, and therefore they have extremely low false positive rates. However, with any new configuration, you should probably observe how the settings you chose are working before you use them to block traffic.

   If you select **More** for one of the signature categories, the **Configure Actions for Signatures** dialog box appears. Its contents are the same as the contents of the **Modify Signatures Object** dialog box, as described in To Configure a Signatures Object.

   If the signatures object has already logged connections, you can click **Logs** to display the syslogs for the category, as described in Logs, Statistics, and Reports.

6. **Select Advanced Protections screen.** On this screen, you choose the advanced protections (also called *security checks* or simply *checks*) that you want to use to protect your web sites and web services. The checks are divided into categories. Which categories are available (and which checks are available within a category) depends on

the profile type that you chose on the **Specify Name** screen. All checks are available for Web 2.0 Application profiles. If you chose that profile type, the **Select advanced protections** screen displays the following categories of security checks:

- Top--level protections (Some checks appear at the top level, not in any category.)
- Data Leak Prevention Protections
- Advanced Form Protections
- URL Protections
- XML Protections

To display the individual checks in a category, click the icon to the left of the category. To apply a security check to your filtered data, select the check box next to the name of the security check. For descriptions of the security checks see Advanced Protections and its subtopics.

7. **Select Advanced Actions screen.** On this screen, you configure the actions for the advanced protections that you have enabled.

   Note: If no advanced protections are enabled, the Wizard skips the **Advanced Actions** screen and goes directly to the **Summary** screen.

   The actions that you can configure are:

   - **Block.** Block connections that match the signature. Disabled by default.
   - **Log.** Log connections that match the signature for later analysis. Enabled by default.
   - **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.
   - **Learn.** Observe traffic to this Web site or Web service, and use connections that repeatedly violate this check to generate recommended exceptions to the check, or new rules for the check. Available only for some checks.

   To enable or disable an action for a check, in the list, select or clear the check box for that action to the right of that check.

   To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check. You can also select a check and, at the bottom of the dialog box, click **Open** to display a dialog box for modifying any of the options for that check. These dialog boxes also vary from check to check. Most of them include a **Checks** tab and a **General** tab. If the check supports relaxations, the **Checks** tab includes an **Add** button, which opens yet another dialog box, in which you can specify a relaxation for the check. A relaxation is a rule for exempting specified traffic from the check.

   For information about the settings available for a check, see the detailed description of that check.

   To review the recommendations generated by the learning engine for a specific check, select that check and then click **Learned Violations** to open the **Manage Learned Rules**

dialog box for that check. For more information on how learning works and how to configure exceptions (relaxations) or deploy learned rules for a check, see Manual Configuration By Using the Configuration Utility under To configure and use the learning feature

To view all logs for a specific check, select that check, and then click **Logs**.

8. **Summary screen.** On this screen, you review your configuration choices to verify that they are what you want. If you want to make changes, you click **Back** until you have returned to the appropriate screen, and make your changes. If the configuration is as you want it, you click **Finish** to save it , and then click **Exit** to close the Application Firewall wizard.

Following are four procedures that show how to perform specific types of configuration by using the Application Firewall wizard.

To configure the Application Firewall: Initial Configuration

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.
3. On the Application Firewall wizard, **Introduction** screen, in the lower right-hand corner, click **Next**.
4. On the **Specify Name** screen, in the **Name** text box, type a name for your new security configuration, and from the **Type** drop-down list, select the type of security configuration. Then, click **Next**.
5. On the **Specify Rule** screen, click **Next** again.

   Note: The default rule, `true`, protects all Web traffic that is sent via your NetScaler appliance or VPX. You can create specific security configurations to protect specific parts of your Web sties or Web applications later.
6. On the **Select Signature Protections** screen, select check boxes to specify the groups of signatures that are appropriate for protecting the content on your protected web sites, and then click **Next**.

   For more information about signatures, see Signatures
7. On the **Select Signature Actions** screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click **Next**.
8. On the **Select Advanced Protections** screen, click **Next** again.

   You typically do not need to configure the security checks during initial configuration.
9. On the **Summary** screen, review your choices to verify that they are what you want. Then, click **Finish**, or click **Back** to return to a previous screen and make changes. When you are finished, click **Exit** to close the Application Firewall wizard.

To configure the Application Firewall: Enabling Blocking for Signatures

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.
3. On the Application Firewall wizard, **Introduction** screen, in the lower right-hand corner, click **Next**.
4. On the **Specify Name** screen, select **Modify Existing Configuration** and, in the Name drop-down list, choose the security configuration that you created during simple configuration, and then click **Next**.
5. In the **Specify Rule** screen, click **Next** again.
6. In the **Select Signature Protections** screen, click **Next** again.
7. In the **Select Signature Actions** screen, enable blocking for your chosen signatures by selecting the **Block** check box to the left of each of those signature.

   For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see Signatures
8. In the **Select advanced protections** screen, click **Next**.
9. On the **Summary** screen, review your choices to verify that they appear correct. Then, click **Finish**, or click **Back** to return to the**Select Signature Actions** screen and make changes. When you are finished, click **Exit** to close the Application Firewall wizard.

To configure the Application Firewall: Enabling and Configuring advanced protection

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.
3. On the Application Firewall wizard, **Introduction** screen, in the lower right-hand corner, click **Next**.
4. On the **Specify Name** screen, select **Modify Existing Configuration** and, in the Name drop-down list, choose the security configuration that you created during simple configuration. Then, click **Next**.
5. On the **Specify Rule** screen, click **Next** again.
6. On the **Select Signature Protections** screen, click **Next**.
7. On the **Select Signature Actions** screen, click **Next** again.
8. On the **Select advanced protections** screen, select the check box beside each security check that you want to enable, and then click **Next**.

   For information about the security checks, see Advanced Protections and its subtopics.
9. On the **Select Deep Actions** screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check, and then click **Next**.

   For general information about the actions, see Advanced Protections and its subtopics. For information about the learning feature, which is available for some security checks, see To configure and use the Learning feature.
10. On the **Summary** screen, review your choices to verify that they appear correct. Then, click **Finish**, or click **Back** to return to the**Select Signature Actions** screen and make changes. When you are finished, click **Exit** to close the Application Firewall wizard.

To configure the Application Firewall: Creating A Policy

The following procedure describes how to use the Application Firewall wizard to create a specialized security configuration to protect only specific content. In this case, you create a new security configuration instead of modifying the initial configuration. This type of security configuration requires a custom rule, so that the policy applies the configuration to only the selected Web traffic.

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.
3. On the Application Firewall wizard, **Introduction** screen, in the lower right-hand corner, click **Next**.

4. On the **Specify Name** screen, type a name for your new security configuration in the **Name** text box, select the type of security configuration from the **Type** drop-down list, and then click **Next**.

5. On the **Specify Rule** screen, enter a rule that matches only that content that you want this Web application to protect, and then click **Next**.

   For a description of policies and policy rules, see Policies.

6. On the **Select Signature Protections** screen, choose the appropriate groups of signatures to protect the content on your protected web sites by selecting the check box beside each group of signatures, and then click **Next**.

   For detailed information about signatures, see Signatures.

7. On the **Select Signature Actions** screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click **Next**. For a detailed description of actions, see , see Signatures.

8. In the **Select Advanced Protections** screen, select the check box beside each security check that you want to enable, and then click **Next**.

   For detailed information about the security checks, see Advanced Protections and its subtopics.

9. In the **Select Advanced Actions** screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check. Then, click **Next**.

   For information about each security check to help you determine which actions to enable, see the Advanced Protections section.

10. On the **Summary** screen, review your choices to verify that they appear correct. Then, click **Finish**, or click **Back** to return to the**Select Signature Actions** screen and make changes. When you are finished, click **Exit** to close the wizard.

Manual Configuration

If you want to bind a profile to a bind point other than Global, you must manually configure the binding. Also, certain security checks require that you either manually enter the necessary exceptions or enable the learning feature to generate the exceptions that your Web sites and Web services need. Some of these tasks cannot be performed by using the application firewall wizard.

If you are familiar with how the application firewall works and prefer manual configuration, you can manually configure a signatures object and a profile, associate the signatures object with the profile, create a policy with a rule that matches the web traffic that you want to configure, and associate the policy with the profile. You then bind the policy to Global, or to a bind point, to put it into effect, and you have created a complete security configuration.

For manual configuration, you can use the configuration utility (a graphical interface) or the command line. Citrix recommends that you use the configuration utility. Not all configuration tasks can be performed at the command line. Certain tasks, such as enabling signatures and reviewing learned data, must be done in the configuration utility. Most other tasks are easier to perform in the configuration utility.

Manual Configuration By Using the Configuration Utility

If you need to configure the Application Firewall feature manually, Citrix recommends that you use the configuration utility. For a description of the configuration utility, see The Application Firewall User Interfaces .

To create and configure a signatures object

Before you can configure the signatures, you must create a new signatures object from the appropriate default signatures object template. Assign the copy a new name, and then configure the copy. You cannot configure or modify the default signatures objects directly. The following procedure provides basic instructions for configuring a signatures object. For more detailed instructions, see Manually Configuring the Signatures Feature If you need to create your own, user defined signatures, see The Signatures Editor.

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select the signatures object that you want to use as a template, and then click **Add**.

   Your choices are:

   - *** Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - *** XPath Injection.** Contains all of the items in the * Default Signatures, and in addition contains the XPath injection rules.
3. In the **Add Signatures Object** dialog box, type a name for your new signatures object, click **OK**, and then click **Close**. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
4. Select the signatures object that you created, and then click **Open**.
5. In the **Modify Signatures Object** dialog box, set the **Display Filter Criteria** options at the left to display the filter items that you want to configure.

   As you modify these options, the results that you specify are displayed in the **Filtered Results** window at the right. For more information about the categories of signatures, see Signatures.
6. In the **Filtered Results** area, configure the settings for a signature by selecting and clearing the appropriate check boxes.
7. When finished, finished, click **Close**.

To create and configure a profile

The main task in configuring a profile is configuring the security checks. If you need additional information before completing some of the steps in this procedure, see Advanced Protections and its subtopics.

1. In the navigation pane, expand **Application Firewall**, and then select **Profiles**.
2. In the details pane, click **Add**.
3. In the **Create Application Firewall Profile** dialog box, type a name for your profile.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
4. Choose the profile type from the drop-down list.

   The profile types are HTML (for HTML-based Web sites), XML (for XML-based Web services) and Web 2.0 (for blogs, RSS feeds, wikis, and other sites that contain both HTML and XML).

   Note: If you are unsure what types of content your profile will protect, you can choose Web 2.0 to make the full range of Application Firewall security checks available to protect your Web site.
5. If you plan to use the learning feature or to enable and configure a large number of advanced protections, select **Advanced**. Otherwise, select **Basic**.

   You probably should use the learning feature if you plan to configure either of the SQL injection checks, either of the cross-site scripting checks, any check that provides protection against Web form attacks, or the cookie consistency check. Unless you include the proper exceptions for your protected Web sites when configuring these checks, they can block legitimate traffic. Anticipating all of the necessary exceptions without creating any that are too broad is difficult. The learning feature makes this task much easier.
6. Click **Create**, and then click **Close**.
7. In the **Profiles** pane, select the profile, and then click **Open**.
8. In the **Configure Application Firewall Profile** dialog box, on the **Security Checks** tab, configure the security checks.

   - To enable or disable an action for a check, in the list, select or clear the check box for that action.
   - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check. You can also select a check and, at the bottom of the dialog box, click **Open** to display a dialog box for modifying any of the options for that check. These dialog boxes also vary from check to check. Most of them include a **Checks** tab and a **General** tab. If the check supports relaxations or user-defined rules, the **Checks** tab includes an **Add** button,

which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.)

If relaxations have already been configured, you can select one and click **Open** to modify it.

- To review learned exceptions or rules for a check, select the check, and then click **Learned Violations**. In the **Manage Learned Rules** dialog box, select each learned exception or rule in turn.
  - To edit the exception or rule, and then add it to the list, click **Edit & Deploy**.
  - To accept the exception or rule without modification, click **Deploy**.
  - To remove the exception or rule from the list, click **Skip**.
- To refresh the list of exceptions or rules to be reviewed, click **Refresh**.
- To open the Learning Visualizer and use it to review learned rules, click **Visualizer**.
- To review the log entries for connections that matched a check, select the check, and then click **Logs**. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see Logs, Statistics, and Reports
- To completely disable a check, in the list, clear all of the check boxes to the right of that check.

9. On the **Settings** tab, configure the profile settings.
   - To associate the profile with the set of signatures that you previously created and configured, under **Common Settings**, choose that set of signatures in the **Signatures** drop-down list.

     Note: You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.
   - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.

     Note: You must first upload the error object that you want to use in the **Imports** pane. For more information about importing error objects, see Imports.
10. If you want to use the learning feature, click **Learning**, and configure the learning settings for the profile, as described in To configure and use the Learning feature.
11. Click **OK** to save your changes and return to the **Profiles** pane.

To configure a relaxation or rule

The following procedure describes how to manually configure a relaxation or rule for a security check that supports user-configured relaxations or rules.

1. Click the **Checks** tab. The **Checks** tab contains a list of relaxations or rules for the security check that you are configuring. The list may be empty, or may have entries, depending on the defaults chosen when you created this profile and whether you have previously added relaxations or rules. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable relaxations or rules.

2. To add a relaxation or rule, click **Add**, flll in the elements in the dialog box that appears, and then click **OK** to save your changes.

   The elements can differ significantly depending upon the specific security check. Following is a complete list of all elements that may appear, and information about each.

   - **Enabled check box.** Select to place this relaxation or rule in active use; clear to deactivate it.
   - **Attachment Content Type.** The Content-Type attribute of an XML attachment. In the text area, enter a regular expression that matches the Content-Type attribute of the XML attachments to allow.
   - **Cookie.** In the text area, enter a PCRE-format regular expression that defines the cookie.
   - **Field Name.** A web form field name element may be labeled Field Name, Form Field, or another similar name. In the text area, enter a PCRE-format regular expression that defines the name of the form field.
   - **Name.** An XML element or attribute name. In the text area, enter a PCRE-format regular expression that defines the name of the element or attribute.
   - **URL.** A URL element may be labeled Action URL, Deny URL, Form Action URL, Form Origin URL, Start URL, or simply URL. In the text area, enter a PCRE-format regular expression that defines the URL.
   - **Format.** The format section contains multiple settings that include list boxes and text boxes. Any of the following can appear:

     - **Type.** Select a field type in the **Type** drop-down list. To add a new field type definition, click **Manage**.
     - **Minimum Length.** Type a positive integer that represents the minimum length in characters if you want to force users to fill in this field. Default: 0 (Allows field to be left blank.)
     - **Maximum length.** To limit the length of data in this field, type a positive integer that represents the maximum length in characters. Default: 65535

Location. Choose the element of the request that your relaxation will apply to from the drop-down list. For HTML security checks, the choices are:

- `FORMFIELD`. Form fields in web forms.
- `HEADER`. Request headers.
- `COOKIE`. Set-Cookie headers.

For XML security checks, the choices are:

- `ELEMENT`. XML element.
- `ATTRIBUE`. XML attribute.

- **Maximum Attachment Size.** The maximum size in bytes allowed for an XML attachment.
- **Comments.** In the text area, type a comment. Optional.

Note: For any element that requires a regular expression, you can type the regular expression, use the **Regex Tokens** menu to insert regular expression elements and symbols directly into the text box, or click **Regex Editor** to open the **Add Regular Expression** dialog box, and use it to construct the expression.

3. To modify a relaxation or rule, select it, and then click **Open**, configure the elements in the dialog box that appears, and then click **OK** to save your changes. See the previous step for a list of elements that may appear in this dialog box.
4. To remove a relaxation or rule, select it, and then click **Remove**.
5. To enable a relaxation or rule, select it, and then click **Enable**.
6. To disable a relaxation or rule, select it, and then click **Disable**.
7. To review learned rules for this check, click **Learning** and perform the steps in To configure and use the Learning feature.
8. Click **OK**.

To configure and use the Learning feature

The learning feature is a repetitive pattern filter that observes activity on a web site or application protected by the application firewall, to determine what constitutes normal activity on that web site or application. It then generates a list of relaxations for supported security checks.. Users normally find it easier to configure relaxations by using the learning feature than by entering relaxations manually.

1. Click the **Learning** tab. At the top of the **Learning** tab is list of the security checks that are available in the current profile and that support the learning feature.

2. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:

   Minimum number threshold. Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1

   Percentage of times threshold. Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
3. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click **Remove All Learned Data**.

   Note: This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.

4. To review learned relaxations for a security check, select that security check, then click **Manage Rules**.

5. In the **Manage Learned Rules** dialog box, choose how you want to review the learned rules.

   - To review the actual learned patterns as displayed in the window, do nothing and proceed to the next step.

   - To review the learned data hierarchically as a branching tree, allowing you to choose general patterns that match many of the learned patterns, click **Visualizer**.
6. If you have chosen to review actual learned patterns, perform the following steps.

   a. Select the first learned relaxation, and choose how to handle it.

      To modify and then accept the relaxation, click **Edit & Deploy**, edit the relaxation regular expression, and then click **OK**.

      To accept the relaxation without modifications, click **Deploy**.

      To remove the relaxation from the list without deploying it, click **Skip**.
   b. Repeat the previous step to review each additional learned relaxation.
7. If you have chosen to use the Learning Visualizer, perform the following steps.

   a. In the branching heirarchical display, select a node that contains a learned pattern, and choose how to handle it.

      The screen area beneath the tree structure, under **Regex of Selected Node**, displays a generalized expression that matches all of the patterns in that node. If you want to display an expression that matches just one of the branches or just one of the leaves, select that branch or leaf.

      -

To modify and then accept the learned relaxation, click **Edit & Deploy**, edit the relaxation regular expression, and then click **OK**.

- To accept the relaxation without modifications, click **Deploy**.
- To remove the modification from the list without deploying it, click **Skip**.

b. Repeat the previous step to review other portions of the display.

c. Click **Close** to return to the **Manage Learned Rules** dialog box.

8. Click **Close** to return to the **Configure Application Firewall Profile** dialog box, **Learning** tab.

To create and configure a policy

1. In the navigation pane, expand **Application Firewall**, then expand **Policies**, and then select **Firewall**.
2. In the details pane, click **Add**.
3. In the **Create Application Firewall Policy** dialog box, **Policy Name** text box, type a name for your new policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
4. Select the profile that you want to associate with this policy from the **Profile** drop-down list. You can create a new profile to associate with your policy by clicking **New**, and you can modify an existing profile by clicking **Modify**.
5. In the **Expression** text area, create a rule for your policy.
   - You can type a rule directly into the text area.
   - You can click **Prefix** to select the first term for your rule, and follow the prompts. See To Create an Application Firewall Rule (Expression) for a complete description of this process.
   - You can click **Add** to open the **Add Expression** dialog box, and use it to construct the rule. See The Add Expression Dialog Box for a complete description of this process.
6. Click **Create**, and then click **Close**.

To bind an Application Firewall policy

1. In the navigation pane, expand **Application Firewall**, select the policy that you created, and then click **Global Bindings**.

2. In the **Bind/Unbind Firewall Policy(s) to Global** dialog box, click **Insert**. A record is inserted into the list of bindings, and in the **Policy Name** column a drop-down list of available policies appears.

3. Select your policy, or click **New Policy** to create a new policy.

4. Click **OK**.

Manual Configuration By Using the NetScaler Command Line

You can configure many application firewall features from the NetScaler command line. There are important exceptions, however. You cannot enable signatures from the command line. There are over 1,000 default signatures in seven categories; the task is simply too complex for the command line interface. You can configure the check actions and parameters for security checks from the command line, but cannot enter manual relaxations. While you can configure the adaptive learning feature and enable learning from the command line, you cannot review learned relaxations or learned rules and approve or skip them. The command line interface is intended for advanced users who are thoroughly familiar with the NetScaler appliance and the application firewall feature.

To manually configure the Application Firewall by using the NetScaler command line, use a telnet or secure shell client of your choice to log on to the NetScaler command line.

To create a profile by using the NetScaler command line

At the command prompt, type the following commands:

- add appfw profile <name> [**-defaults** (**basic** | **advanced**)]
- set appfw profile <name> **-type** (**HTML** | **XML** | **HTML XML**)
- save ns config

Example

The following example adds a profile named `pr-basic`, with basic defaults, and assigns a profile type of `HTML`. This is the appropriate initial configuration for a profile to protect an HTML Web site.

```
add appfw profile pr-basic -defaults basic
set appfw profile pr-basic -type HTML
save ns config
```

Parameters for Creating a Profile

name (Profile Name)

A name for the profile. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

defaults (Defaults)

You can choose one of two default configurations when you create a profile: Basic or Advanced. A profile created with basic defaults should protect most Web sites while requiring little additional configuration. A profile created with advanced defaults is intended to protect more complex Web sites requiring additional configuration. You can modify either type of default configuration.

type (Profile Type)

The type of content that the profile will protect. There are three types of profile: **HTML** (HTML), **XML** (XML), or **Web 2.0** (HTML XML). If you are unsure what types of content your profile will protect, you can specify Web 2.0 to make the full range of Application Firewall security checks available to protect your Web site.

To configure a profile by using the NetScaler command line

At the command prompt, type the following commands:

- set appfw profile <name> <arg1> [<arg2> ...] where <arg1> represents a parameter and <arg2> represents either another parameter or the value to assign to the parameter represented by <arg1>. For descriptions of the parameters to use when configuring specific security checks, see Advanced Protections and its subtopics. For descriptions of the other parameters, see "Parameters for Creating a Profile."
- save ns config

Example

The following example shows how to configure an HTML profile created with basic defaults to begin protecting a simple HTML-based Web site. This example turns on logging and maintenance of statistics for most security checks, but enables blocking only for those checks that have extremely low false positive rates and require no special configuration. It also turns on transformation of unsafe HTML and unsafe SQL, which prevents attacks but does not block requests to your Web sites. With logging and statistics enabled, you can later review the logs to determine whether to enable blocking for a specific security check.

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
```

```
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
```

To create and configure a policy

At the NetScaler command prompt, type the following commands:

- add appfw policy <name> <rule> <profile>
- save ns config

Example

The following example adds a policy named `pl-blog`, with a rule that intercepts all traffic to or from the host `blog.example.com`, and associates that policy with the profile `pr-blog`. This is an appropriate policy to protect a blog hosted on a specific hostname.

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
```

Parameters for creating and configuring a policy

name

A name for your policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

rule

A policy rule, or expression, in NetScaler expressions language. For a short description and some useful examples of Application Firewall rules, see . For a complete description of the NetScaler expressions language, see the *Citrix NetScaler Policy Configuration and Reference Guide*

profile

The name of the profile that you previously created.

To bind an Application Firewall policy

At the NetScaler command prompt, type the following commands:

- bind appfw global <policy> <priority>
- save ns config

Example

The following example binds the policy named `pl-blog` and assigns it a priority of `10`.

```
bind appfw global pl-blog 10
save ns config
```

PCRE Character Encoding Format

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your application firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your application firewall configuration as a URL, form field name, or Safe Object expression. They include:

Upper-ASCII characters. Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.

Double-Byte characters. Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.

ASCII control characters. Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, “English US,” the application firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The application firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The application firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
-

Japanese (SJIS). The application firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.

- **Japanese (EUC-JP).** The application firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.

- **Korean (EUC-KR).** The application firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.

- **Turkish (ISO-8859-9).** The application firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.

- **Unicode (UTF-8).** The application firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the application firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the application firewall configuration is “\xNN” for ASCII characters; “\xNN\xNN” for non-ASCII characters used in English, Russian, and Turkish; and “\xNN\xNN\xNN” for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an application firewall regular expression as a UTF-8 character, you would type \x21. If you want to include an á, you would type \xC3\xA1.

Note: Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character’s UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as \x20 to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the application firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters.

  Actual URL: `http://www.josénuñez.com`

  Encoded URL: `^http://www[.]jos\xC3\xA9nu\xC3\xB1ez[.]com$`

- Another URL containing extended ASCII characters.

  Actual URL: `http://www.example.de/trömso.html`

  Encoded URL: `^http://www[.]example[.]de/tr\xC3\xB6mso[.]html$`

- A form field name containing extended ASCII characters.

  Actual Name: `nome_do_usuário`

  Encoded Name: `^nome_do_usu\xC3\xA1rio$`

- A safe object expression containing extended ASCII characters.

  Unencoded Expression `[A-Z]{3,6}¥[1-9][0-9]{6,6}`

  Encoded Expression: `[A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}`

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

http://www.utf8-chartable.de/unicode-utf8-table.pl

For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Signatures

The application firewall signatures function provides specific, configurable rules that protect your web sites against known attacks. A signature represents a pattern that is a component of a known attack on an operating system, a web server, a web site, an XML-based web service, or any other server that is connected to a web site or web service. A signature can consist of a literal string or a PCRE-compliant regular expression.

To specify how the application firewall is to use signatures, you configure a signatures object, which specifies the signatures to apply to your traffic and the actions to be taken when the signatures match the traffic. A signatures object also contains the SQL injection and cross-site scripting patterns, and may also contain XPath injection patterns. These patterns are not actually signatures but are used by some of the advanced protection checks. The SQL Injection and Cross-Site Scripting patterns contain the SQL special symbols and keywords, the cross-site scripting allowed tags and attributes, and the denied patterns for the HTML and XML SQL Injection and Cross-Site Scripting checks. The XPath injection patterns contain the XPath (*XML Path Language*) denied patterns.

Note: If you use the wizard to configure signatures, the signatures object is created automatically.

The application firewall examines requests to your protected web sites and web services to determine whether a request matches a signature. Matching requests are handled as you specify when configuring the Signatures actions. By default, matching requests are logged so that you can examine them later. If you enabled blocking, the application firewall displays an error page or error object. If you enabled statistics, the application firewall also includes the request in the statistics that it maintains about requests that match an application firewall signature or security check.

If you want to configure signatures manually, you must create a signatures object from a template. There are two default templates that you can use: the ***Default Signatures** template and the ***Xpath Injection** template. The ***Default Signatures** template contains 1,209 signatures, in addition to the complete list of SQL injection and cross-site scripting allowed and denied patterns. The ***Xpath Injection** template contains all of those, and in addition contains 57 Xpath keywords and special strings.

Once you have created a signatures object, you can configure all parts of it, including the signatures rules, the XML SQL Injection and Cross-Site Scripting rules, and the Xpath injection rules. You can also import signatures from a supported vulnerability scanner such as Cenzic, and you can manually create and modify your own custom signatures in the signatures editor. You can also add new SQL injection, cross-site scripting, and XPath injection patterns, modify existing patterns, and remove patterns.

Important: Regardless of whether you use the wizard for initial configuration or configure your signatures object manually, you should regularly apply the Citrix updates to keep your signatures current.

Manually Configuring the Signatures Feature

To use signatures to protect your web sites, you must review the rules, and enable and configure the ones that you want to apply. The rules are disabled by default. Citrix recommends that you enable all rules that are applicable to the type of content that your web site uses.

To manually configure the signatures feature, use a browser to connect to the configuration utility. Then, create a signatures object from a template or an existing signatures object and configure the new object.

Note: The following procedures do not address adding user-defined signatures to a signatures object. To create your own signatures, see The Signatures Editor.

To Create a Signatures Object from a Template

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select the signatures object that you want to use as a template.

   Your choices are:

   - *** Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - *** XPath Injection.** Contains everything that is in the *** Default Signatures** template, and also contains the XPath injection rules.
3. Click **Add**.
4. In the **Add Signatures Object** dialog box, type a name for your new signatures object, and then click **OK**. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
5. Proceed with configuring the new signatures object, or click **Close**. To proceed with configuring the new object, see steps 3 and following of the procedure described in "To Configure a Signatures Object." The contents of your **Add Signatures Object** dialog box are the same as those of the **Modify Signatures Object** dialog box.

To Configure a Signatures Object

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select the signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, set the **Display Filter Criteria** options at the left to display the filter items that you want to configure.

   As you modify these options, the results that you requested are displayed in the Filtered Results window at the right.

   - To display only selected categories of signatures, check or clear the appropriate signature-category check boxes. The signature categories are:

Name	Type of Attack that this Signature Protects Against
cgi	CGI scripts. Includes Perl and UNIX shell scripts.
client	Browsers and other clients.
coldfusion	Web sites that use the Adobe Systems ColdFusion application server.
frontpage	Web sites that use Microsoft's FrontPage server.
iis	Web sites that use the Microsoft Internet Information Server (IIS).
misc	Miscellaneous attacks.
php	Web sites that use PHP

   - To display only signatures that have specific check actions enabled, select the **ON** check box for each of those actions, clear the **ON** check boxes for the other actions, and clear all of the **OFF** check boxes. To display only signatures that have a specific check action disabled, select their respective **OFF** check boxes and clear all of the **ON** check boxes. To display signatures regardless of whether they have a check action enabled or disabled, select or clear both the **ON** and the **OFF** check boxes for that action. The check actions are:

Criterion	Description
Enabled	The signature is enabled. The application firewall checks only for signatures that are enabled when it processes traffic.
Block	Connections that match this signature are blocked.
Log	A log entry is produced for any connection that matches this signature.
Stats	The application firewall includes any connection that matches this signature in the statistics that it generates for that check.

   - To display only signatures that contain a specific string, type the string into the text box under the filter criteria, and then click **Search**.
   - To reset all display filter criteria to the default settings and display all signatures, click **Show All**.

4. For information about a specific signature, select the signature, and then click the blue double arrow to the left of the **Category** field. The **Signature Rule Vulnerability Detail** message box appears. It contains information about the purpose of the signature and provides links to external web-based information about the vulnerability or vulnerabilities that this signature addresses. To access an external link, click the blue double arrow to the left of the description of that link.

5. Configure the settings for a signature by selecting the appropriate check boxes.

6. If you want to add a local signature rule to the signatures object, or modify an existing local signature rule, see The Signatures Editor.

7. If you have no need for SQL injection, cross-site scripting, or Xpath injection patterns, click **OK**, and then click **Close**. Otherwise, in the lower left-hand corne of the details pane, click **Manage SQL/XSS Patterns**.

8. In the **Manage SQL/XSS Patterns** dialog box, **Filtered Results** window, navigate to the pattern category and pattern that you want to configure. For information about the SQL injection patterns, see HTML SQL Injection Check. For information about the cross-site scripting patterns, see HTML Cross-Site Scripting Check.

9. To add a new pattern:

    a. Select the branch to which you want to add the new pattern.

    b. Click the **Add** button directly below the lower section of the **Filtered Results** window.

    c. In the **Create Signature Item** dialog box, fill in the **Element** text box with the pattern that you want to add. If you are adding a transformation pattern to the transformrules branch, under **Elements**, fill in the **From** text box with the pattern that you want to change and the **To** text box with the pattern to which you want to change the previous pattern.

    d. Click **OK**.

10. To modify an existing pattern:

    a. In the **Filtered Results** window, select the branch that contains the pattern that you want to modify.

    b. In the detail window beneath the **Filtered Results** window, select the pattern that you want to modify.

    c. Click **Modify**.

    d. In the **Modify Signature Item** dialog box, **Element** text box, modify the pattern. If you are modifying a transformation pattern, you can modify either or both patterns under **Elements**, in the **From** and the **To** text boxes.

    e. Click **OK**.

11. To remove a pattern, select the pattern that you want to remove, then click the **Remove** button below the details pane beneath the **Filtered Results** window. When prompted, confirm your choice by clicking **Close**.

12. To add the patterns category to the XSS branch:

a. Select the branch to which you want to add the patterns category.

b. Click the **Add** button directly below the **Filtered Results** window.

Note: Currently you can add only one category, named patterns, to the XSS branch, so after you click **Add**, you must accept the default choice, which is `patterns`.

c. Click **OK**.

13. To remove a branch, select that branch, and then click the **Remove** button directly below the **Filtered Results** window. When prompted, confirm your choice by clicking **OK**.

Note: If you remove a default branch, you remove all of the patterns in that branch. Doing so can disable the security checks that use that information.

14. When you are finished modifying the SQL injection, cross-site scripting, and XPath injection patterns, click **OK**, and then click **Close** to return to the **Modify Signatures Object** dialog box.

15. Click **OK** at any point to save your changes, and when you are finished configuring the signatures object, click **Close**.

Updating a Signatures Object

Citrix regularly updates the signatures for the Application Firewall. You should regularly update the signatures on your Application Firewall to ensure that your Application Firewall is using the most current list. Ask your Citrix representative or Citrix reseller for the URL to access the updates.

To Update a Signatures Object

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select the signatures object that you want to update, and then click **Update**.
3. In the **Update Signatures Object** dialog box, choose one of the following options.
   - **Import from URL.** Choose this option if you download signature updates from a web URL.
   - **Import from Local File.** Choose this option if you import signature updates from a file on your local hard drive, network hard drive, or other storage device.
4. In the text area, type the URL, or type or browse to the local file.
5. Click **Update**. The update file is imported, and the **Update Signatures** dialog box changes to a format nearly identical to that of the **Modify Signatures Object** dialog box, which is described in To Configure a Signatures Object. The **Update Signatures Object** dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
6. Review and configure the new and modified signatures.
7. When you are finished, click **OK**, and then click **Close**.

Updating a Signatures Object from a Supported Vulnerability Scanning Tool

You can add and update signatures generated by a supported vulnerability scanning tool, such as Cenzic, to the signatures in a signatures object. Before you update signatures, you must export them from the vulnerability scanning tool to a file.

To Import and Update Signatures from a Vulnerability Scanning Tool

1. In the navigation pane, expand **Application Firewall**, and then click **Signatures**.
2. In the details pane, select the signatures object that you want to update, and then click **Update**.
3. In the **Update Signatures Object** dialog box, on the **External Format** tab, choose one of the following options.
   - **Import from URL.** Choose this option if you download signature updates from a Web URL.
   - **Import from Local File.** Choose this option if you import signature updates from a file on your local or a network hard drive or other storage device.
4. In the text area, type the URL, or browse or type the path to the local file.
5. Click **Update**. The update file is imported, and the **Update Signatures** dialog box changes to a format nearly identical to that of the **Modify Signatures Object** dialog box, which is described in To Configure a Signatures Object. The **Update Signatures Object** dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
6. Review and configure the new and modified signatures.
7. When you are finished, click **OK**, and then click **Close**.

The Signatures Editor

You can use the signatures editor, which is available in the configuration utility, to add a new user-defined (*local*) signature rule to an existing signatures object, or to modify a previously configured local signature rule. Except that it is defined by the user (you), a local signature rule has the same attributes as a default signature rule from Citrix, and it functions in the same way. You enable or disable it, and configure the signature actions for it, just as you do for a default signature.

Add a local rule if you need to protect your web sites and services from a known attack that the existing signatures do not match. For example, you might discover a new type of attack and determine its characteristics by examining the logs on your web server, or you might obtain third-party information about a new type of attack.

At the heart of a signature rule are the rule *patterns*, which collectively describe the characteristics of the attack that the rule is designed to match. Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting patterns.

You might want to modify a signature rule by adding a new pattern or modifying an existing pattern to match an attack. For example, you might find out about changes to an attack, or you might determine a better pattern by examining the logs on your web server, or from third-party information.

Parameters for the Signature Editor

Actions

The **Enabled**, **Block**, **Log**, and **Stats** check boxes. Select or clear the check boxes as appropriate to configure what the Application Firewall does when the signature rule is matched.

- To enable the signature rule, select **Enabled**.
- To enable blocking for the signature rule, select **Block**.
- To enable logging for the signature rule, select **Log**.
- To enable maintaining of statistics for the signature rule, select **Stats**.

Category

The category in which the signature rule is included. Assigning a signature to a category enables you to configure it and other signatures in that category as a group, which can significantly speed up the configuration process when you want to configure many signature rules in the same way.

LogString

A brief string (a few words or a short sentence) that describes the attack that this signature rule matches. This string is used when logging a match to the NetScaler log.

Comment

A longer description of the attack and the signature rule, with information that another user who is configuring this signature object would need to know. Optional.

Patterns

The patterns that describe the characteristics of the attack that this signature rule should match. Patterns can be fixed strings, PCRE-format regular expressions, or the built-in SQL injection or cross-site scripting patterns. See Signature Rule Patterns for more information.

Note: If a signature rule has more then one pattern, a match does not occur unless all of the patterns match the data to which the rule is compared. If either of two patterns can define an attack, but both would not necessarily be present, you should create two signature rules.

To add a signature by using the Signatures Editor

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select the signatures object that you want to edit, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results window**, click **Add**.
4. In the **Add Local Signature Rule** dialog box, configure the actions for a signature by selecting the appropriate check boxes.
5. Choose a category for the new signature rule from the **Category** drop-down list.

   You can also create a new category by clicking the icon to the right of the list and using the **Add Signature Rule Category dialog** box to add a new category to the list, The rule you are modifying is automatically added to the new category. For instructions, see To add a signature rule category.
6. In the **LogString** text box, type a brief description of the signature rule to be used in the logs.
7. In the **Comment** text box, type a comment. (Optional)
8. In the **Patterns** list, click **Add**, and in the **Create New Signature Rule Pattern** dialog box add one or more patterns for your signature rule. For instructions, see Signature Rule Patterns.
9. Click **OK**.

To modify a signature by using the Signatures Editor

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select the signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results window**, select that signature that you want to configure, and then click **Open**.
4. In the **Modify Local Signature Rule** dialog box, configure the actions for a signature by selecting the appropriate check boxes.
5. Choose a category for the new signature rule from the **Category** drop-down list.

   You can also create a new category by clicking the icon to the right of the list, and using the **Add Signature Rule Category dialog** box to add a new category to the list. The rule you are modifying is automatically added to the new category. To add a signature rule category.
6. In the **LogString** text box, modify the description of the signature rule to be used in the logs.
7. In the **Comment** text box, type a comment. (Optional)
8. In the **Patterns** list, add or edit a pattern.

   - To add a pattern, click **Add**. In the **Create New Signature Rule Pattern** dialog box, add one or more patterns for your signature rule, and then click **OK**.
   - To edit a pattern, select the pattern, and then click **Open**. In the **Edit Signature Rule Pattern** dialog box, modify the pattern, and then click **OK**.

   For more information about adding or editing patterns, see Signature Rule Patterns.
9. Click **OK**.

To add a signature rule category

Putting signature rules into a category enables you to configure the actions for a group of signatures instead of for each individual signature. You might want to do so for the following reasons:

- **Ease of selection.** For example, assume that all of signature rules in a particular group protect against attacks on a specific type of web server software or technology. If your protected web sites use that software or technology, you want to enable them all. If they do not, you do not want to enable any of them.
- **Ease of initial configuration.** It is easiest to set defaults for a group of signatures as a category, instead of one-by-one. You can then make any changes to individual signatures as needed.
- **Ease of ongoing configuration.** It is easier to configure signatures if you can display only those that meet specific criteria, such as belonging to a specific category.

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen, beneath the Filtered Results window, click **Add**.
4. In the **Add Local Signature Rule** dialog box, click the icon to the right of the **Category** drop-down list.
5. In the **Add Signature Rule Category** dialog box, **New Category** text box, type a name for your new signature category. The name can consist of from one to 64 characters and must start with an alphanumeric character.
6. Click **OK**.

Signature Rule Patterns

You can add a new pattern to a signature rule or modify an existing pattern of a signature rule to specify a string or expression that characterizes an aspect of the attack that the signature matches. To determine which patterns an attack exhibits, you can examine the logs on your web server, use a tool to observe connection data in real time, or obtain the string or expression from a third-party report about the attack.

Caution: Any new pattern that you add to a signature rule is in an AND relationship with the existing patterns. Do not add a new pattern to an existing signature rule if you do not want a potential attack to have to match all of the patterns in order to match the signature.

Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting pattern. Before you attempt to add a pattern that is based on a regular expression, you should make sure that you understand PCRE-format regular expressions. PCRE expressions are complex and powerful; if you do not understand how they work, you can unintentionally create a pattern that matches something that you did not want (a *false positive*) or that fails to match something that you did want (a *false negative*).

If you are not already familiar with PCRE-format regular expressions, you can use the following resources to learn the basics, or for help with some specific issue:

- *"Mastering Regular Expressions"*, Third Edition. Copyright (c) 2006 by Jeffrey Friedl. O'Reilly Media, ISBN: 9780596528126
- *"Regular Expressions Cookbook"*. Copyright (c) 2009 by Jan Goyvaerts and Steven Levithan. O'Reilly Media, ISBN: 9780596520687
- **PCRE Man page/Specification** (text/official): http://www.pcre.org/pcre.txt
- **PCRE Man Page/Specification** (html/gammon.edu.au): http://www.gammon.com.au/pcre/index.html
- **Wikipedia PCRE entry**: http://en.wikipedia.org/wiki/PCRE
- **PCRE Mailing List** (run by exim.org): http://lists.exim.org/mailman/listinfo/pcre-dev

If you need to encode non-ASCII characters in a PCRE-format regular expression, the NetScaler platform supports encoding of hexadecimal UTF-8 codes. For more information, see PCRE Character Encoding Format.

To configure a signature rule pattern

1. In the navigation pane, expand **Application Firewall**, and then select **Signatures**.

2. In the details pane, select that signatures object that you want to configure, and then click **Open**.

3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results** window, either click **Add** to create a signature rule, or select an existing signature rule and click **Open.**

   Note: You can modify only signature rules that you added. You cannot modify the default signature rules.

   Depending on your action, either the **Add Local Signature Rule** or the **Modify Local Signature Rule** dialog box appears. Both dialog boxes have the same contents.

4. Under the **Patterns** window in the dialog box that you opened, either click **Add** to add a new pattern, or select an existing pattern from the list beneath the **Add** button and click **Open**. Depending on your action, either the **Create New Signature Rule Pattern** or the **Edit Signature Rule Pattern** dialog box appears. Both dialog boxes have the same contents.

5. In the **Location** area, define the elements to examine with this pattern,. The **Location** area describes what elements of the HTTP request or response to examine for this pattern. As you choose a value from the **Area** drop-down list, the remaining parts of the **Location** area change interactively. Following are all configuration items that might appear in this section.

   Area

   Drop-down list of elements that describe a particular portion of the HTTP connection. The choices are as follows:

   - **HTTP_ANY**. All parts of the HTTP connection.
   - **HTTP_COOKIE**. Cookies in the HTTP headers.
   - **HTTP_FORM_FIELD**. Form fields and their contents.
   - **HTTP_HEADER**. The contents of the HTTP header.
   - **HTTP_RAW_HEADER**. The entire HTTP header.
   - **HTTP_METHOD**. The HTTP method.
   - **HTTP_POST_BODY**. HTTP post bodies containing web form data.
   - **HTTP_STATUS_CODE**. The HTTP status code.
   - **HTTP_STATUS_MESSAGE**. The HTTP status message.
   - **HTTP_URL**. The contents of a URL in the HTTP headers or body.
   - **HTTP_RAW_URL**. An entire URL in the HTTP headers or body.

URL

Examines any URLs found in the elements specified by the **Area** setting.

- To enable, select the **Enabled** check box.
- To search for a literal string in a URL, type the string in the text area.
- To search for a pattern defined by a regular expression, select the **Is Regular Expression** check box, and then type the regular expression in the text area. Use the **Regex Tokens** to insert common regular expression elements at the cursor, or the **Regex Editor** for more assistance in constructing the regular expression that you want.

Field Name

Examines any form field names found in the elements specified by the **Area** selection.

- To enable, select the **Enabled** check box.
- To search for a literal string in a form field, type the string in the text area.
- To search for a pattern defined by a regular expression, select the **Is Regular Expression** check box, and then type the regular expression in the text area. Use the **Regex Tokens** to insert common regular expression elements at the cursor, or the **Regex Editor** for more assistance in constructing the regular expression that you want.

6. In the **Pattern** area, define the pattern. A pattern is a literal string or PCRE-format regular expression that defines the pattern that you want to match. The **Pattern** area contains the following elements:

Match

A drop-down list of four elements that describes the type of pattern you are using.

- **Literal**. A literal string.
- **PCRE**. A PCRE-format regular expression.

  NOTE: When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for other types of patterns.

- **Injection**. Directs the application firewall to look for injected SQL in the specified location. The **Pattern** window disappears, because the application firewall already has the patterns for SQL injection.

  NOTE: If you choose **Injection**, a warning appears, telling you that the **Area** will be reset to **HTTP_FORM_FIELD**. Click **Yes** to accept this change, or **No** to reject it and retain your current **Area** setting.

- **CrossSiteScripting**. Directs the application firewall to look for cross-site scripts in the specified location. The **Pattern** window disappears, because the application firewall already has the patterns for cross-site scripts.

Pattern Window (unlabeled)

In this window, type the pattern that you want to match.

- **Literal.** Simply type the string you want to search for.
- **PCRE.** Type the regular expression in the text area. Use the **Regex Tokens** to insert common regular expression elements at the cursor, or the **Regex Editor** for more assistance in constructing the regular expression that you want.
- **Offset (under *More*>>).** The number of characters to skip over before starting to match on this pattern. You use this field to start examining a string at some point other than the first character.
- **Depth (under *More*>>).** How many characters from the starting point to examine for matches. You use this field to limit searches of a large string to a specific number of characters.
- **Min-Length (under *More*>>).** The minimum match length. Any match that contains fewer than the specified number of characters is not considered to be a valid match.
- **Max-Length (under *More*>>).** The maximum match length. Any match that contains more than the specified number of characters is not considered to be a valid match.
- **Pattern Type (under *More*>>).** A list box with one choice, **fastmatch**. You can use fastmatch only for a literal pattern, to improve performance.

7. Click **OK**.
8. Repeat the previous four steps to add or modify additional patterns.
9. When finished adding or modifying patterns, click **OK** to save your changes and return to the **Signatures** pane.

Caution: Until you click **OK** in the **Add Local Signature Rule** or **Modify Local Signature Rule** dialog box, your changes are not saved. Do not close either of these dialog boxes without clicking **OK** unless you want to discard your changes.

Advanced Protections

The application firewall advanced protections (*security checks*) are a set of filters designed to catch complex or unknown attacks on your protected web sites and web services. The security checks use heuristics, positive security, and other techniques to detect attacks that may not be detected by signatures alone. You configure the security checks by creating and configuring an application firewall *profile*, which is a collection of user-defined settings that tell the application firewall which security checks to use and how to handle a request or response that fails a security check. A profile is associated with a *signatures object* and with a *policy* to create a security configuration.

The application firewall provides twenty security checks, which differ widely in the types of attacks that they target and how complex they are to configure. The security checks are organized into the following categories:

- **Common security checks.** Checks that apply to any aspect of web security that either does not involve content or is equally applicable to all types of content.
- **HTML security checks.** Checks that examine HTML requests and responses. These checks apply to HTML-based web sites and to the HTML portions of Web 2.0 sites, which contain mixed HTML and XML content.
- **XML security checks.** Checks that examine XML requests and responses. These checks apply to XML-based web services and to the XML portions of Web 2.0 sites.

The security checks protect against a wide range of types of attack, including attacks on operation system and web server software vulnerabilities, SQL database vulnerabilities, errors in the design and coding of web sites and web services, and failures to secure sites that host or can access sensitive information.

All security checks have a set of configuration options, the check actions, which control how the application firewall handles a connection that matches a check. Three check actions are available for all security checks. They are:

- **Block.** Block connections that match the signature. Disabled by default.
- **Log.** Log connections that match the signature, for later analysis. Enabled by default.
- **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.

A fourth check action, **Learn**, is available for more than half of the check actions. It observes traffic to a protected Web site or web service and uses connections that repeatedly violate the security check to generate recommended exceptions (*relaxations*) to the check, or new rules for the check. In addition to the check actions, certain security checks have parameters that control the rules that the check uses to determine which connections violate that check, or that configure the application firewall's response to connections that violate the check. These parameters are different for each check, and they are described in the documentation for each check.

To configure security checks, you can use the application firewall wizard, as described in The Application Firewall Wizard, or you can configure the security checks manually, as described in Manual Configuration By Using the Configuration Utility and Manual Configuration By Using the Configuration Utility. Some tasks, such as manually entering relaxations or rules or reviewing learned data, can be done only by using the configuration utility, not the command line. Using the wizard is usually best configuration method, but in some cases manual configuration might be easier if you are thoroughly familiar with it and simply want to adjust the configuration for a single security check.

Regardless of which method you use to configure the security checks, each security check requires that certain tasks be performed. Many checks require that you specify exceptions (*relaxations*) to prevent blocking of legitimate traffic before you enable blocking for that security check. You can do this manually, by observing the log entries after a certain amount of traffic has been filtered and then creating the necessary exceptions. However, it is usually much easier to enable the learning feature and let it observe the traffic and recommend the necessary exceptions.

Top-Level Advanced Protections

Four of the advanced protections are especially effective against common types of Web attacks, and are therefore more commonly used than any of the others. They are:

HTML Cross-Site Scripting. Examines requests and responses for scripts that attempt to access or modify content on a different Web site than the one on which the script is located. When this check finds such a script, it either renders the script harmless before forwarding the request or response to its destination, or it blocks the connection.

HTML SQL Injection. Examines requests that contain form field data for attempts to inject SQL commands into an SQL database. When this check detects injected SQL code, it either blocks the request or renders the injected SQL code harmless before forwarding the request to the Web server.

Note: If both of the following conditions apply to your configuration, you should make certain that your Application Firewall is correctly configured:

- If you enable the HTML Cross-Site Scripting check or the HTML SQL Injection check (or both), and
- Your protected Web sites accept file uploads or contain Web forms that can contain large POST body data.

For more information about configuring the Application Firewall to handle this case, see Configuring the Application Firewall

- **Buffer Overflow.** Examines requests to detect attempts to cause a buffer overflow on the Web server.
- **Cookie Consistency.** Examines cookies returned with user requests to verify that they match the cookies your Web server set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the Web server.

The Buffer Overflow check is simple; you can usually enable blocking for it immediately. The other three top-level checks are considerably more complex and require configuration before you can safely use them to block traffic. Citrix strongly recommends that, rather than attempting to configure these checks manually, you enable the learning feature and allow it to generate the necessary exceptions.

HTML Cross-Site Scripting Check

The HTML Cross-Site Scripting check examines both the headers and the POST bodies of user requests for possible cross-site scripting attacks. If it finds a cross-site script, it either modifies (*transforms*) the request to render the attack harmless, or blocks the request.

To prevent misuse of the scripts on your protected web sites to breach security on your web sites, the HTML Cross-Site Scripting check blocks scripts that violate the *same origin rule*, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of Javascript-enhanced web content that violates the same origin rule. If you enable the HTML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity.

If you use the wizard or the configuration utility, in the **Modify HTML Cross-Site Scripting Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, **Learn**, and **Statistics** actions, and in addition the following parameters:

Transform. If enabled, the application firewall makes the following changes to requests that match the HTML Cross-Site Scripting check:

- Left angle bracket (<) to HTML character entity equivalent (<)
- Right angle bracket (>) to HTML character entity equivalent (>)

This ensures that browsers do not interpret unsafe html tags, such as `<script>`, and thereby execute malicious code. If you enable both request-header checking and transformation, any special characters found in request headers are also modified as described above. If scripts on your protected web site contain cross-site scripting features, but your web site does not rely upon those scripts to operate correctly, you can safely disable blocking and enable transformation. This configuration ensures that no legitimate web traffic is blocked, while stopping any potential cross-site scripting attacks.

Check complete URLs. If checking of complete URLs is enabled, the application firewall examines entire URLs for HTML cross-site scripting attacks instead of checking just the query portions of URLs.

Check Request headers. If Request header checking is enabled, the application firewall examines the headers of requests for HTML cross-site scripting attacks, instead of just URLs.

If you use the command-line interface, you can enter the following commands to configure the HTML Cross-Site Scripting Check:

- set appfw profile <profileName> **-crossSiteScriptingAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <profileName> **-crossSiteScriptingTransformUnsafeHTML** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-crossSiteScriptingCheckCompleteURLs** ([**ON**] | [**OFF**])

To specify relaxations for the HTML Cross-Site Scripting check, you must use the configuration utility. On the **Checks** tab of the **Modify HTML Cross-Site Scripting Check** dialog box, click **Add** to open the **Add HTML Cross-Site Scripting Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify HTML Cross-Site Scripting Check Relaxation**dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of HTML Cross-Site Scripting check relaxations:

Web Form Field Expressions

Logon Fields. The following expression exempts all fields beginning with the string `logon_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{2,15}$
```

Name Fields. The following expression exempts form fields with names beginning with `Name_` followed by a string beginning with a letter or number and consisting of from one to twenty letters, numbers, or the apostrophe or hyphen symbol:

```
^Name_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$
```

Name Fields (Special Characters). If your web site has Turkish-speaking customers whose first names may contain special characters, you might have a form field that begins with the string Türkçe-Name_ on their logon page. In addition, the customers may use the same special characters in their names. The special characters in both of these strings must be represented as encoded UTF-8 strings. The following expression exempts form fields beginning with Türkçe-Name_ and containing Turkish special characters:

```
^T\xC3\xBCrk\xC3\xA7e-Name_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$
```

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Session-ID Fields. The following expression exempts all fields beginning with the string `sessionid-` followed by a ten-digit number:

```
^sessionid-[0-9]{10,10}$
```

URL Expressions

- **URLs using Javascript.** You can use a single expression to exempt all URLs that end with a filename that follows a specified pattern. The following expression exempts all URLs that end with the string `query_` followed by a string of upper- and lower-case

letters or numbers that is at least two characters long and no more than forty characters long, and ending with the string `.js`:

```
query_[0-9A-Za-z]{2,40}[.]js$
```

- **URLs containing a Specified String.** You can use an expression to exempt all URLs that contain a specific string. The following expression exempts all URLs that contain the string `prodinfo`:

  ```
  ^https?://(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f]+[.])+[a-z]{2,6}/[^<>?]*\?prodinfo[^<>?]*$
  ```

  In the above expression , each character class has been grouped with the string `\\x[0-9A-Fa-f][0-9A-Fa-f]`, which matches all properly constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (\\) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If you included only one backslash, the application firewall would interpret the following left square bracket ([) as a literal character instead of as the opening of a character class, which would break the expression.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML Cross-Site Scripting check would otherwise have blocked.

HTML SQL Injection Check

The HTML SQL Injection check provides special defenses against injection of unauthorized SQL code that might break security. It examines both the headers and the POST bodies of requests for injected SQL code. If the application firewall detects unauthorized SQL code in a user request, it either transforms the request, to render the SQL code inactive, or blocks the request.

Many web applications have web forms that use SQL to communicate with relational database servers. Often, the scripts that pass web form information to the database do not validate the information provided by the user before sending it to the database. Malicious code or a hacker can use the insecure web form to send SQL commands to the web server.

Note: To prevent blocking of legitimate requests, this check ignores cookies that were set by the server, even if they contain elements that the Cookie Consistency Check would otherwise block.

If you use the wizard or the configuration utility, in the **Modify Cookie Consistency Check** dialog box, on the **General** tab you can enable or disable the Block, Log, Statistics, and Learn actions, and the following actions:

- **Transform.** Make the following changes to requests that match the HTML SQL Injection check:
  - Single straight quote (') to double straight quote (")
  - Backslash (\) to double backslash (\\).
  - Semicolon (;) is dropped completely.

  These three characters (*special strings*) are necessary to issue commands to an SQL server. Unless an SQL command is prefaced with a special string, most SQL servers ignore that command. For this reason, the changes that the application firewall performs when transformation is enabled prevent an attacker from injecting active SQL. After these changes are made, it is safe to forward the request to your protected web site. When web forms on your protected web site may legitimately contain SQL special strings, but the web form does not rely upon the special strings to operate correctly, you can disable blocking and enable transformation to prevent blocking of legitimate web form data without reducing the protection that the application firewall provides to your protected web sites.

  Note: You normally enable either transformation or blocking, but not both. If you have blocking enabled, enabling transformation is redundant because the application firewall already blocks access to requests that contain injected SQL.

- **Restrict checks to fields containing SQL special characters.** If you configure the application firewall to check only fields that contain SQL special strings, the application firewall skips web form fields that do not contain special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this parameter can significantly reduce the load on the application firewall and speed up processing without placing your protected web sites at risk.

- **SQL comments handling.** By default, the application firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if it is preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure the application firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:
  - **ANSI.** Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.
  - **Nested.** Skip nested SQL comments, which are normally used by Microsoft SQL Server.
  - **ANSI/Nested.** Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.

    Caution: In most cases, you should not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they may indicate an attempt to breach security on that server.
  - **Check all Comments.** Check the entire request for injected SQL, without skipping anything. The default setting.
- **Check Request headers.** Examine the headers of requests for HTML SQL Injection attacks, instead of just URLs.

  Caution: If you enable both request header checking and transformation, any SQL special characters found in headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;), so enabling both Request header checking and transformation simultaneously may cause errors.

If you use the command-line interface, you can enter the following commands to configure the HTML SQL Injection Check:

- set appfw profile <profileName> **-SQLInjectionAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <profileName> **-SQLInjectionTransformSpecialChars** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-SQLInjectionOnlyCheckFieldsWithSQLChars** ([**ON**]|[**OFF**])
- set appfw profile <profileName> **-SQLInjectionParseComments** ([**checkall**] | [**ansi**|**nested**] | [**ansinested**])

To specify relaxations for the HTML SQL Injection check, you must use the configuration utility. On the **Checks** tab of the **Modify HTML SQL Injection Check** dialog box, click **Add** to open the **Add HTML SQL Injection Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify HTML SQL Injection Check Relaxation** dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of HTML SQL Injection check relaxations:

Web Form Field Name Expressions

Logon Fields. The following expression exempts all fields beginning with the string `logon_` followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{2,15}$
```

Name Fields. The following expression exempts form fields with names beginning with `Name_` followed by a string beginning with a letter or number and consisting of from one to twenty letters, numbers, or the apostrophe or hyphen symbol:

```
^Name_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$
```

Name Fields (Special Characters). If your web site has Turkish-speaking customers whose first names may contain special characters, you might have a form field that begins with the string Türkçe-Name_ on their logon page. In addition, the customers may use the same special characters in their names. The special characters in both of these strings must be represented as encoded UTF-8 strings. The following expression exempts form fields beginning with Türkçe-Name_ and containing Turkish special characters:

```
^T\xC3\xBCrk\xC3\xA7e-Name_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$
```

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Session-ID Fields. The following expression exempts all fields beginning with the string `sessionid-` followed by a ten-digit number:

```
^sessionid-[0-9]{10,10}$
```

Action URL Expressions

URLs using Javascript. You can use a single expression to exempt all URLs that end with a filename that follows a specified pattern. The following expression exempts all URLs that end with the string `query_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than forty characters long, and that end with the string `.js`:

```
query_[0-9A-Za-z]{2,40}[.]js$
```

URLs containing a Specified String. You can use an expression to exempt all URLs that contain a specific string. The following expression exempts all URLs that contain the string `prodinfo`:

```
^https?://(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])(([0-9A-Za-z_-]|\\x[0-9A-Fa-f]
[0-9A-Fa-f]+[.])+[a-z]{2,6}/[^<>?]*\?prodinfo[^<>?]*$
```

In the expression above, each character class has been grouped with the string `\\x[0-9A-Fa-f][0-9A-Fa-f]`, which matches all properly constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (\\) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If you included only one backslash, the application firewall would interpret the following left square bracket ([) as a literal character instead of as the opening of a character class, which would break the expression.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML SQL Injection check would otherwise have blocked.

Buffer Overflow Check

The Buffer Overflow check detects attempts to cause a buffer overflow on the web server. If the application firewall detects a URL, cookie or header longer than the specified maximum length in a request, it blocks that request because it might be an attempt to cause a buffer overflow.

The Buffer Overflow check prevents attacks against insecure operating-system or web-server software that can crash or behave unpredictably when it receives a data string that is larger than it can handle. Proper programming techniques prevent buffer overflows by checking incoming data and either rejecting or truncating overlong strings. Many programs, however, do not check all incoming data and are therefore vulnerable to buffer overflows. This issue especially affects older versions of web-server software and operating systems, many of which are still in use.

If you use the wizard or the configuration utility, in the **Modify Buffer Overflow Check** dialog box, on the **General** tab you can enable or disable the Block, Log, and Statistics actions. On the **Checks** tab, you can set the following parameters:

- **Maximum URL Length.** The maximum length the application firewall allows in a requested URL. Requests with longer URLs are blocked. Possible Values: 0-65536. Default: 1024
- **Maximum Cookie Length.** The maximum length the application firewall allows for an individual cookie in a request. Longer cookies are stripped from requests before those requests are forwarded to your protected web server. Possible Values: 0-65536. Default: 4096
- **Maximum Header Length.** The maximum length the application firewall allows for HTTP headers. Requests with longer headers are blocked. Possible Values: 0-65536. Default: 4096

If you use the command-line interface, you can add the following Buffer Overflow Check arguments to the set appfwl profile <profileName> command:

- **-bufferOverflowAction [block] [log| [stats]**
- **-bufferOverflowMaxURLLength** <positiveInteger>
- **-bufferOverflowMaxCookieLength** <positiveInteger>
- **-bufferOverflowMaxHeaderLength** <positiveInteger>

Cookie Consistency Check

The Cookie Consistency check examines cookies returned by users, to verify that they match the cookies that your web site set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the web server. You can also configure the Cookie Consistency check to transform all of the server cookies that it processes, by encrypting the cookies, proxying the cookies, or adding flags to the cookies. This check applies to requests and responses.

An attacker would normally modify a cookie to gain access to sensitive private information by posing as a previously authenticated user, or to cause a buffer overflow. The Buffer Overflow check protects against attempts to cause a buffer overflow by using a very long cookie. The Cookie Consistency check focuses on the first scenario.

If you use the wizard or the configuration utility, in the **Modify Cookie Consistency Check** dialog box, on the **General** tab you can enable or disable the following actions:

- Block
- Log
- Learn
- Statistics
- Transform. If enabled, the Transform action modifies all cookies as specified in the following settings:
    - **Encrypt Server Cookies.** Encrypt cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list, before forwarding the response to the client. Encrypted cookies are decrypted when the client sends a subsequent request, and the decrypted cookies are reinserted into the request before it is forwarded to the protected web server. Specify one of the following types of encryption:
        - **None.** Do not encrypt or decrypt cookies. The default.
        - **Decrypt only.** Decrypt encrypted cookies only. Do not encrypt cookies.
        - **Encrypt session only.** Encrypt session cookies only. Do not encrypt persistent cookies. Decrypt any encrypted cookies.
        - **Encrypt all.** Encrypt both session and persistent cookies. Decrypt any encrypted cookies.

          Note: When encrypting cookies, the application firewall adds the **HttpOnly** flag to the cookie. This flag prevents scripts from accessing and parsing the cookie. The flag therefore prevents a script-based virus or trojan from accessing a decrypted cookie and using that information to breach security. This is done regardless of the **Flags to Add in Cookies** parameter settings, which are handled independently of the **Encrypt Server Cookies** parameter

settings.

- **Proxy Server Cookies.** Proxy all non-persistent (*session*) cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list. Cookies are proxied by using the existing application firewall session cookie. The application firewall strips session cookies set by the protected web server and saves them locally before forwarding the response to the client. When the client sends a subsequent request, the application firewall reinserts the session cookies into the request before forwarding it to the protected web server. Specify one of the following settings

  - **None.** Do not proxy cookies. The default.

  - **Session only.** Proxy session cookies only. Do not proxy persistent cookies.

    Note: If you disable cookie proxying after having enabled it (set this value to **None** after it was set to **Session only**), cookie proxying is maintained for sessions that were established before you disabled it. You can therefore safely disable this feature while the application firewall is processing user sessions.
- **Flags to Add in Cookies**. Add flags to cookies during transformation. Specify one of the following settings:

  - **None.** Do not add flags to cookies. The default.

  - **HTTP only.** Add the HttpOnly flag to all cookies. Browsers that support the HttpOnly flag do not allow scripts to access cookies that have this flag set.

  - **Secure**. Add the Secure flag to cookies that are to be sent only over an SSL connection. Browsers that support the Secure flag do not send the flagged cookies over an insecure connection.

  - **All.** Add the HttpOnly flag to all cookies, and the Secure flag to cookies that are to be sent only over an SSL connection.

If you use the command-line interface, you can enter the following commands to configure the Cookie Consistency Check:

- set appfw profile <profileName> **-cookieConsistencyAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]

- set appfw profile <profileName> **-cookieTransforms** ([**ON**] | [**OFF**])

- set appfw profile <profileName> **-cookieEncryption** ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])

- set appfw profile <profileName> **-cookieProxying** ([**none**] | [**sessionOnly**])

- set appfw profile <profileName> **-addCookieFlags** ([**none**] | [**httpOnly**] | [**secure**] | [**all**])

To specify relaxations for the Cookie Consistency check, you must use the configuration utility. On the **Checks** tab of the **Modify Cookie Consistency Check** dialog box, click **Add** to open the **Add Cookie Consistency Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify Cookie Consistency Check Relaxation**dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of Cookie Consistency check relaxations:

- **Logon Fields.** The following expression exempts all form fields beginning with the string `logon_` followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

  ```
  ^logon_[0-9A-Za-z]{2,15}$
  ```

- **Logon Fields (special characters).** The following expression exempts all form fields beginning with the string `türkçe-logon_` followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

  ```
  ^t\xC3\xBCrk\xC3\xA7e-logon_[0-9A-Za-z]{2,15}$
  ```

  Note: The special characters in that string must be represented as encoded UTF-8 strings. See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

- **Arbitrary strings.** Allow cookies that contain the string `sc-item_`, followed by the ID of an item that the user has added to his shopping cart ([0-9A-Za-z]+), a second underscore (_), and finally the number of these items he wants ([1-9][0-9]?), to be user-modifiable:

  ```
  ^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
  ```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

Data Leak Prevention Checks

The data-leak-prevention checks filter responses to prevent leaks of sensitive information, such as credit card numbers and social security numbers, to unauthorized recipients.

Credit Card Check

The Credit Card check provides special handling for credit card numbers. A web application does not usually send a credit card number in a response to a user request, even when the user supplies a credit card number in the request. The application firewall examines web server responses, including headers, for credit card numbers. If it finds a credit card number in the response, and the administrator has not configured it to allow credit card numbers to be sent, it responds in one of two ways:

- It blocks the response.
- It replaces all but the final group of digits in the credit card with x's. For example, a credit card number of 9876-5432-1234-5678 would be rendered xxxx-xxxx-xxxx-5678.

The Credit Card check prevents attackers from exploiting a security flaw in your web server software or on your web site to obtain credit card numbers of your customers. If your web sites do not have access to credit card information, you do not need to configure this check. If you have a shopping cart or other application that can access credit card numbers, or your web sites have access to database servers that contain credit card numbers, you should configure protection for each type of credit card that you accept.

Note: A web site that does not access an SQL database usually does not have access to sensitive private information such as credit card numbers.

If you use the wizard or the configuration utility, in the **Credit Card Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions, and the following actions:

- **X-Out.** Mask any credit card number detected in a response by replacing each digit, except the digits in the final group, with the letter "X."
- **Maximum credit cards allowed per page.** Allow up to the specified number of credit card numbers per page in responses without masking the credit card numbers or blocking the response. The Maximum is set to zero (0) by default. Web pages do not usually contain unmasked credit card numbers, but occasionally a web page might legitimately contain a credit card number or even a list of credit card numbers. To allow one or more credit card numbers to appear in a web page before masking the numbers or blocking the response, change the value in the "Maximum credit cards allowed per page" text box to the number of credit cards that you want to allow.

To configure the types of credit cards to be protected, in the **Modify Credit Card Check** dialog box, select each credit card type that you want to protect, and then click **Protect**. If you want to cancel protection for a credit card type, select that credit card type and then click **Unprotect**. You can hold down your Shift or Ctrl key while choosing credit card types, and then enable or disable several credit card types at once by clicking the **Protect** or **Unprotect** button while multiple credit card types are selected.

If you use the command-line interface, you can enter the following commands to configure the Credit Card Check:

- set appfw profile <profileName> **-creditCardAction** [**block**] [**log**] [**stats**] [**none**]

- set appfw profile <profileName> **-creditCard** (**VISA** | **MASTERCARD** | **DISCOVER** | **AMEX** | **JCB** | **DINERSCLUB**)
- set appfw profile <profileName> **-creditCardMaxAllowed** <integer>
- set appfw profile <profileName> **-creditCardXOut** ([**ON**] | [**OFF**])

Safe Object Check

The Safe Object check provides user-configurable protection for sensitive business information, such as customer numbers, order numbers, and country-specific or region-specific telephone numbers or postal codes. A user-defined regular expression or custom plug-in tells the application firewall the format of this information and defines the rules to be used to protect it. If a string in a user request matches a safe object definition, the application firewall either blocks the response, masks the protected information, or removes the protected information from the response before sending it to the user, depending on how you configured that particular safe object rule.

The Safe Object check prevents attackers from exploiting a security flaw in your web server software or on your web site to obtain sensitive private information, such as company credit card numbers or social security numbers. If your web sites do not have access to these types of information, you do not need to configure this check. If you have a shopping cart or other application that can access such information, or your web sites have access to database servers that contain such information, you should configure protection for each type of sensitive private information that you handle and store.

Note: A web site that does not access an SQL database usually does not have access to sensitive private information.

The **Safe Object Check** dialog box is unlike that for any other check. Each safe object expression that you create is the equivalent of a separate security check, similar to the **Credit Card** check, for that type of information. If you use the wizard or the configuration utility, you add a new expression by clicking **Add** and configuring the expression in the **Add Safe Object** dialog box. You modify an existing expression by selecting it, then clicking **Open**, and then configuring the expression in the **Modify Safe Object** dialog box.

In the **Safe Object** dialog box for each safe object expression, you can configure the following:

- **Safe Object Name.** A name for your new safe object. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 255 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.
- **Actions.** Enable or disable the **Block**, **Log**, and **Statistics** actions, and the following actions:
  - **X-Out.** Mask any information that matches the safe object expression, by replacing each character with the letter “X”.
  - **Remove.** Remove any information that matches the safe object expression.
- **Regular Expression.** Enter a PCRE-compatible regular expression that defines the safe object. You can create the regular expression in one of three ways: by typing the regular expression directly into the text box, by using the **Regex Tokens** menu to enter regular expression elements and symbols directly into the text box, or by opening the **Regular Expressions Editor** and using it to construct the expression. The regular expression must consist of ASCII characters only. Do not cut and paste characters that are not part of the basic 128-character ASCII set. If you want to include non-ASCII

characters, you must manually type those characters in PCRE hexadecimal character encoding format.

Note: Do not use start anchors (^) at the beginning of Safe Object expressions, or end anchors ($) at the end of Safe Object expressions. These PCRE entities are not supported in Safe Object expressions, and if used, will cause your expression not to match what it was intended to match.

- **Maximum Match Length.** Enter a positive integer that represents the maximum length of the string that you want to match. For example, if you want to match U.S. social security numbers, enter the number eleven (11) in this field. That allows your regular expression to match a string with nine numerals and two hyphens. If you want to match California driver's license numbers, enter the number eight (8).

  Caution: If you do not enter a maximum match length in this field, the application firewall uses a default value of one (1) when filtering for strings that match your safe object expressions. As a result, most safe object expressions fail to match their target strings.

You cannot use the command-line interface to configure the Safe Object check. You must configure it by using either the application firewall wizard or the configuration utility.

Following are examples of Safe Object check regular expressions:

- Look for strings that appear to be U.S. social security numbers, which consist of three numerals (the first of which must not be zero), followed by a hyphen, followed by two more numerals, followed by a second hyphen, and ending with a string of four more numerals:

  [1-9][0-9]{2,2}-[0-9]{2,2}-[0-9]{4,4}

- Look for strings that appear to be California driver's license IDs, which start with a letter and are followed by a string of exactly seven numerals:

  [A-Za-z][0-9]{7,7}

- Look for strings that appear to be Example Manufacturing customer IDs which, consist of a string of five hexadecimal characters (all the numerals and the letters A through F), followed by a hyphen, followed by a three-letter code, followed by a second hyphen, and ending with a string of ten numerals:

  [0-9A-Fa-f]{5,5}-[A-Za-z]{3,3}-[0-9]{10,10}

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write to ensure that they define exactly the type of string you want to add as a safe object definition, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you did not want or expect, such as blocking access to web content that you did not intend to block.

Advanced Form Protection Checks

The advanced Form Protection checks examine web form data to prevent attackers from compromising your system by modifying the web forms on your web sites or sending unexpected types and quantities of data to your web site in a form.

Field Formats Check

The Field Formats check verifies the data that users send to your web sites in a web form. It examines both the length and type of data to ensure that it is appropriate for the form field in which it appears. If the application firewall detects inappropriate web form data in a user request, it blocks the request. This check applies to HTML requests only. It does not apply to XML requests.

By preventing an attacker from sending inappropriate web form data to your web site, the Field Formats check prevents certain types of attacks on your web site and database servers. For example, if a particular field expects the user to enter a phone number, the Field Formats check examines the user's response to ensure that the data matches the format for a phone number. If a particular field expects a first name, the Field Formats check ensures that the data in that field is of a type and length appropriate for a first name. It does the same thing for each form field that you configure it to protect.

The Field Formats check provides a different type of protection than does the Form Field Consistency check. The Form Field Consistency check verifies that the structure of the web forms returned by users is intact, that data format restrictions configured in the HTML are respected, and that data in hidden fields has not been modified. It can do this without any specific knowledge about your web forms other than what it derives from the web form itself. The Field Formats check verifies that the data in each form field matches the specific formatting restrictions that you configured manually, or that the learning feature generated and you approved. In other words, the Form Field Consistency check enforces general web form security, while the Field Formats check enforces the specific rules that you set for your web forms.

Before it can protect your web forms, the Field Formats check requires that you configure the application firewall to recognize the type and length of data expected in each form field on each web form that you want to protect.

If you use the wizard or the configuration utility, in the **Modify Field Formats Check** dialog box, on the **General** tab you can enable or disable the Block, Log, Learn, and Statistics actions and the following parameters:

- **Field Type.** Assign a default field type to form fields in web forms that do not have a field type. This parameter is not set by default. You can assign any field type that is defined on your application firewall as the default field type.

  Caution: If you set a restrictive default field type and do not disable blocking until you are certain that the field types assigned to your form fields are correct, users may be unable to use your web forms.

- **Minimum Length.** The default minimum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 0 by default, which allows the user to leave the field blank. Any higher setting forces users to fill in the field.

- **Maximum Length.** The default maximum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 65535 by default.

If you use the command-line interface, you can enter the following commands to configure the Field Formats Check:

- set appfw profile <profileName> **-fieldFormatAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <profileName> **-defaultFieldFormatType** <string>
- set appfw profile <profileName> **-defaultFieldFormatMinLength** <integer>
- set appfw profile <profileName> **-defaultFieldFormatMaxLength** <integer>

To specify relaxations for the Field Formats check, you must use the configuration utility. On the **Checks** tab of the **Modify Field Formats Check** dialog box, click **Add** to open the **Add Field Formats Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify Field Formats Check Relaxation**dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of Field Formats check relaxations:

Choose form fields with the name FirstName:

^FirstName$

Choose form fields with names that begin with Name_ and are followed by a string beginning with a letter or number and consisting of from one to twenty letters, numbers, or the apostrophe or hyphen symbol:

^Name_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$

Choose form fields with names that begin with Türkçe-FirstName_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

^T\xC3\xBCrk\xC3\xA7e-FirstName_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

^[0-9A-Za-z]*Num[0-9A-Za-z]*$

Form Field Consistency Check

The Form Field Consistency check examines the web forms returned by users of your web site, and verifies that the web form was not modified inappropriately by the client. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The Form Field Consistency check prevents clients from making unauthorized changes to the structure of the web forms on your web site when they are filling out a web form and submitting data by using that form. It also ensures that the data a user submits meets the HTML restrictions for length and type, and that data in hidden fields is not modified. This prevents an attacker from tampering with a web form and using the modified form to gain unauthorized access to web site, redirect the output of a contact form that uses an insecure script and thereby send unsolicited bulk email, or exploit a vulnerability in your web server software to gain control of the web server or the underlying operating system. Web forms are a weak link on many web sites and attract a wide range of attacks.

The Form Field Consistency check verifies all of the following:

- If a field is sent to the user, the check ensures that it is returned by the user.
- The check enforces HTML field lengths and types.

  Note: The Form Field Consistency check enforces HTML restrictions on data type and length but does not otherwise validate the data in web forms. You can use the Field Formats check to set up rules that validate data returned in specific form fields on your web forms
- If your web server does not send a field to the user, the check does not allow the user to add that field and return data in it.
- If a field is a read-only or hidden field, the check verifies that the data has not changed.
- If a field is a list box or radio button field, the check verifies that the data in the response corresponds to one of the values in that field.

If a web form returned by a user violates one or more of the Form Field consistency checks, and you have not configured the application firewall to allow that web form to violate the Form Field Consistency checks, the request is blocked.

If you use the wizard or the configuration utility, in the **Modify Form Field Consistency Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, **Learn**, and **Statistics** actions.

If you use the command-line interface, you can enter the following command to configure the Form Field Consistency Check:

- set appfw profile <profileName> **-fieldConsistencyAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]

To specify relaxations for the Form Field Consistency check, you must use the configuration utility. On the **Checks** tab of the **Modify Form Field Consistency Check** dialog box, click **Add** to open the **Add Form Field Consistency Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify Form Field Consistency Check Relaxation**dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of Form Field Consistency check relaxations:

Form Field Names

Choose form fields with the name UserType:

^UserType$

Choose form fields with names that begin with UserType_ and are followed by a string that begins with a letter or number and consists of from one to twenty-one letters, numbers, or the apostrophe or hyphen symbol:

^UserType_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$

Choose form fields with names that begin with Türkçe-UserType_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$

Note: See [PCRE Character Encoding Format] ... for a complete description of supported special characters and how to encode them properly.

Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

^[0-9A-Za-z]*Num[0-9A-Za-z]*$

Form Field Action URLs

- Choose URLs beginning with http://www.example.com/search.pl? and containing any string after the query except for a new query:

  ^http://www[.]example[.]com/search[.]pl\?[^?]*$

- Choose URLs that begin with http://www.example-español.com and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

  ^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

- Choose all URLs that contain the string /search.cgi?:

  ^[^?<>]*/search[.]cgi\?[^?<>]*$

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

CSRF Form Tagging Check

The CSRF Form Tagging check tags each web form sent by a protected web site to users with a unique and unpredictable FormID, and then examines the web forms returned by users to ensure that the supplied FormID is correct. This check protects against Cross Site Request Forgery (CSRF) attacks. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The CSRF Form Tagging check prevents attackers from using their own web forms to send high volume form responses with data to your protected web sites. This check requires relatively little CPU processing capacity compared to certain other security checks that analyze web forms in depth. It is therefore able to handle high volume attacks without seriously degrading the performance of the protected web site or the application firewall itself.

Before you enable the CSRF Form Tagging check, you should be aware of the following:

- You need to enable form tagging. The CSRF check depends on form tagging and does not work without it.
- You should disable the Citrix NetScaler Integrated Caching feature for all web pages containing forms that are protected by that profile. The Integrated Caching feature and CSRF form tagging are not compatible.
- You should consider enabling Referer checking. Referer checking is part of the Start URL check, but it prevents cross-site request forgeries, not Start URL violations. Referer checking also puts less load on the CPU than does the CSRF Form Tagging check. If a request violates Referer checking, it is immediately blocked, so the CSRF Form Tagging check is not invoked.
- The CSRF Form Tagging check does not work with web forms that use different domains in the form-origin URL and form-action URL. For example, CSRF Form Tagging cannot protect a web form with a form-origin URL of http://www.example.com/ and a form action URL of http://www.example.org/form.pl, because example.com and example.org are different domains.

If you use the wizard or the configuration utility, in the **Modify CSRF Form Tagging Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions.

If you use the command-line interface, you can enter the following command to configure the CSRF Form Tagging Check:

- set appfw profile <profileName> **-fieldConsistencyAction** [**block**] [**log**] [**stats**] [**none**]

To specify relaxations for the CSRF Form Tagging check, you must use the configuration utility. On the **Checks** tab of the **Modify CSRF Form Tagging Check** dialog box, click **Add** to open the **Add CSRF Form Tagging Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify CSRF Form Tagging Check Relaxation** dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of CSRF Form Tagging check relaxations:

Note: The following expressions are URL expressions that can be used in both the Form Origin URL and Form Action URL roles.

Choose URLs beginning with http://www.example.com/search.pl? and containing any string after the query, except for a new query:

```
^http://www[.]example[.]com/search[.]pl\?[^?]*$
```

Choose URLs that begin with http://www.example-español.com and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
```

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Choose all URLs that contain the string /search.cgi?:

```
^[^?<>]*/search[.]cgi\?[^?<>]*$
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the check would otherwise have blocked.

Deny URL Check

The Deny URL check examines and blocks connections to URLs that are commonly accessed by hackers and malicious code. This check contains a list of URLs that are common targets of hackers or malicious code and that rarely if ever appear in legitimate requests. You can also add URLs or URL patterns to the list. The Deny URL check prevents attacks against various security weaknesses known to exist in web server software or on many web sites.

The Deny URL check takes priority over the Start URL check, and thus denies malicious connection attempts even when a Start URL relaxation would normally allow a request to proceed.

If you use the wizard or the configuration utility, in the **Modify Deny URL Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions.

If you use the command-line interface, you can enter the following command to configure the Deny URL Check:

- set appfw profile <profileName> **-denyURLAction** [**block**] [**log**] [**stats**] [**none**]

To create and configure your own deny URLs, you must use the configuration utility. On the **Checks** tab of the **Modify Deny URL Check** dialog box, click **Add** to open the **Add Deny URL** dialog box, or select an existing user-defined deny URL and click **Open** to open the **Modify Deny URL**dialog box. Either dialog box provides the same options for creating and configuring a deny URL, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of Deny URL expressions:

Do not allow users to access the image server at `images.example.com` directly:

^http://images[.]example[.]com$

Do not allow users to access CGI (`.cgi`) or PERL (`.pl`) scripts directly:

^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$

Here is the same deny URL, modified to support non-ASCII characters:

^http://www[.]example[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL or pattern that you want to block, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results that you do not want, such as

blocking access to web content that you did not intend to block.

URL Protection Checks

The URL Protection checks examine request URLs to prevent attackers from aggressively attempting to access multiple URLs (*forceful browsing*) or using a URL to trigger a known security vulnerability in web server software or web site scripts.

Start URL Check

The Start URL check examines the URLs in incoming requests and blocks the connection attempt if the URL does not meet the specified criteria. To meet the criteria, the URL must match an entry in the Start URL list, unless the Enforce URL Closure parameter is enabled. If you enable this parameter, a user who clicks a link on your Web site is connected to the target of that link.

The primary purpose of the Start URL check is to prevent repeated attempts to access random URLs on a Web site, (*forceful browsing*). Forceful browsing can be used to trigger a buffer overflow, find content that users were not intended to access directly, or find a back door into secure areas of your Web server.

If you use the wizard or the configuration utility, in the **Modify Start URL Check** dialog box, on the **General** tab you can enable or disable **Block**, **Log**, **Statistics**, and **Learn** actions, and the following parameters:

- **Enforce URL Closure.** Allow users to access any web page on your web site by clicking a hyperlink on any other page on your web site. Users can navigate to any page on your web site that can be reached from the home page or any designated start page by clicking hyperlinks.

  Note: The URL closure feature allows any query string to be appended to and sent with the action URL of a web form submitted by using the HTTP GET method. If your protected web sites use forms to access an SQL database, make sure that you have the SQL injection check enabled and properly configured.

- **Validate Referer Header.** Verify that the Referer header in a request that contains web form data from your protected web site instead of another web site. This action verifies that your web site, not an outside attacker, is the source of the web form. Doing so protects against cross-site request forgeries (CSRF) without requiring form tagging, which is more CPU-intensive than header checks. The application firewall can handle the HTTP Referer header in one of the following three ways, depending on which option you select in the drop-down list:

  - **Off.** Do not validate the Referer header.

  - **If-Present.** Validate the Referer header if a Referer header exists. If an invalid Referer header is found, the request generates a referer-header violation. If no Referer header exists, the request does not generate a referer-header violation. This option enables the application firewall to perform Referer header validation on requests that contain a Referer header, but not block requests from users whose browsers do not set the Referer header or who use web proxies or filters that remove that header.

  - **Always.** Always validate the Referer header. If there is no Referer header, or if the Referer header is invalid, the request generates a referer-header violation.

  Note: Although the referer header check and Start URL security check share the same action settings, it is possible to violate the referer header check without violating the Start URL check. The difference is visible in the logs, which log referer header check violations separately from Start URL check violations.

One Start URL setting, **Exempt Closure URLs from Security Checks**, is not configured in the **Modify Start URL Check** dialog box, but in the **Settings** tab of the **Configure Application Firewall Profile** dialog box. If enabled, this setting directs the application firewall not to run further security checks on URLs that meet the URL Closure criteria.

If you use the command-line interface, you can enter the following commands to configure the Start URL Check:

- set appfw profile <profileName> **-startURLAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <profileName> **-startURLClosure** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-exemptClosureURLsFromSecurityChecks** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-RefererHeaderCheck** ([**none**] | [**if-present**] | [**always**])

To specify relaxations for the Start URL check, you must use the configuration utility. On the **Checks** tab of the **Modify Start URL Check** dialog box, click **Add** to open the **Add Start URL Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify Start URL Check Relaxation**dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of Start URL check relaxations:

- Allow users to access the home page at `www.example.com`:

  ^http://www[.]example[.]com$

- Allow users to access all static HTML (`.htm` and `.html`), server-parsed HTML (`.htp` and `.shtml`), PHP (`.php`), and Microsoft ASP (`.asp`) format web pages at `www.example.com`:

  ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
  [0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$

- Allow users to access web pages with pathnames or file names that contain non-ASCII characters at `www.example-español.com`:

  ^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x
  ([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)

  Note: In the above expression, each character class has been grouped with the string `\\x[0-9A-Fa-f][0-9A-Fa-f]`, which matches all properly-constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (\\) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If

you included only one backslash, the application firewall would instead interpret the following left square bracket ([) as a literal character instead of the opening of a character class, which would break the expression. See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

- Allow users to access all GIF (`.gif`), JPEG (`.jpg` and `.jpeg`), and PNG (`.png`) format graphics at `www.example.com`:

  ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
  [0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$

- Allow users to access CGI (`.cgi`) and PERL (`.pl`) scripts, but only in the CGI-BIN directory:

  ^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$

- Allow users to access Microsoft Office and other document files in the `docsarchive` directory:

  ^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_-.]*[.](doc|xls|pdf|ppt)$

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions that you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the Start URL check would otherwise have blocked.

Deny URL Check

The Deny URL check examines and blocks connections to URLs that are commonly accessed by hackers and malicious code. This check contains a list of URLs that are common targets of hackers or malicious code and that rarely if ever appear in legitimate requests. You can also add URLs or URL patterns to the list. The Deny URL check prevents attacks against various security weaknesses known to exist in web server software or on many web sites.

The Deny URL check takes priority over the Start URL check, and thus denies malicious connection attempts even when a Start URL relaxation would normally allow a request to proceed.

If you use the wizard or the configuration utility, in the **Modify Deny URL Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions.

If you use the command-line interface, you can enter the following command to configure the Deny URL Check:

- set appfw profile <profileName> **-denyURLAction** [**block**] [**log**] [**stats**] [**none**]

To create and configure your own deny URLs, you must use the configuration utility. On the **Checks** tab of the **Modify Deny URL Check** dialog box, click **Add** to open the **Add Deny URL** dialog box, or select an existing user-defined deny URL and click **Open** to open the **Modify Deny URL**dialog box. Either dialog box provides the same options for creating and configuring a deny URL, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of Deny URL expressions:

Do not allow users to access the image server at `images.example.com` directly:

^http://images[.]example[.]com$

Do not allow users to access CGI (`.cgi`) or PERL (`.pl`) scripts directly:

^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$

Here is the same deny URL, modified to support non-ASCII characters:

^http://www[.]example[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL or pattern that you want to block, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results that you do not want, such as

blocking access to web content that you did not intend to block.

XML Protection Checks

The XML Protection checks examine requests for XML-based attacks of all types.

Caution: The XML security checks apply only to content that is sent with an HTTP content-type header of text/xml. If the content-type header is missing, or is set to a different value, all XML security checks are bypassed. If you plan to protect XML or Web 2.0 web applications, the webmasters of each web server that hosts those applications should ensure that the proper HTTP content-type header is sent.

XML Format Check

The XML Format check examines the XML format of incoming requests and blocks those requests that are not well formed or that do not meet the criteria in the XML specification for properly-formed XML documents. Some of those criteria are:

- An XML document must contain only properly-encoded Unicode characters that match the Unicode specification.
- No special XML syntax characters—such as < , > and &—can be included in the document except when used in XML markup.
- All begin, end, and empty-element tags must be correctly nested, with none missing or overlapping.
- XML element tags are case-sensitive. All beginning and end tags must match exactly.
- A single root element must contain all the other elements in the XML document.

A document that does not meet the criteria for well-formed XML does not meet the definition of an XML document. Strictly speaking, it is not XML. However, not all XML applications and web services enforce the XML well-formed standard, and not all handle poorly-formed or invalid XML correctly. Inappropriate handling of a poorly-formed XML document can cause security breaches. The purpose of the XML Format check is to prevent a malicious user from using a poorly-formed XML request to breach security on your XML application or web service.

If you use the wizard or the configuration utility, in the **Modify XML Format Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions.

If you use the command-line interface, you can enter the following command to configure the XML Format Check:

- set appfw profile <profileName> **-xmlFormatAction** [**block**] [**log**] [**stats**] [**none**]

You cannot configure exceptions to the XML Format check. You can only enable or disable it.

XML Denial-of-Service Check

The XML Denial of Service (*XML DoS* or *XDoS*) check examines incoming XML requests to determine whether they match the characteristics of a denial-of-service (DoS) attack, and blocks those requests that do. The purpose of the XML DoS check is to prevent an attacker from using XML requests to launch a denial-of-service attack on your web server or web site.

If you use the wizard or the configuration utility, in the **Modify XML Denial-of-Service Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, **Statistics**, and **Learn** actions:

If you use the command-line interface, you can enter the following command to configure the XML Denial-of-Service check:

- set appfw profile <profileName> **-xmlDoSAction** [**block**] [**log**] [**learn**] [**stats**] [**none**]

To configure individual XML Denial-of-Service rules, you must use the configuration utility. On the **Checks** tab of the **Modify XML Denial-of-Service Check** dialog box, select a rule and click **Open** to open the **Modify XML Denial-of-Service** dialog box for that rule. The individual dialog boxes differ for the different rules but are extremely simple. Some only allow you to enable or disable the rule; others allow you to modify a number by typing a new value in a text box.

The individual XML Denial-of-Service rules are:

Maximum Element Depth

Restrict the maximum number of nested levels in each individual element to 256. If this rule is enabled, and the application firewall detects an XML request with an element that has more than the maximum number of allowed levels, it blocks the request. You can modify the maximum number of levels to any value from one (1) to 65,535.

Maximum Element Name Length

Restrict the maximum length of each element name to 128 characters. This includes the name within the expanded namespace, which includes the XML path and element name in the following format:

{http://prefix.example.com/path/}target_page.xml

The user can modify the maximum name length to any value between one (1) character and 65,535.

Maximum # Elements

Restrict the maximum number of any one type of element per XML document to 65,535. You can modify the maximum number of elements to any value between one (1) and 65,535.

Maximum # Element Children

Restrict the maximum number of children (including other elements, character information, and comments) each individual element is allowed to have to 65,535. You can modify the maximum number of element children to any value between one (1) and 65,535.

Maximum # Attributes

Restrict the maximum number of attributes each individual element is allowed to have to 256. You can modify the maximum number of attributes to any value between one (1) and 256.

Maximum Attribute Name Length

Restrict the maximum length of each attribute name to 128 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

Maximum Attribute Value Length

Restrict the maximum length of each attribute value to 2048 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

Maximum Character Data Length

Restrict the maximum character data length for each element to 65,535. You can modify the length to any value between one (1) and 65,535.

Maximum File Size

Restrict the size of each file to 20 MB. You can modify the maximum file size to any value.

Minimum File Size

Require that each file be at least 9 bytes in length. You can modify the minimum file size to any positive integer representing a number of bytes.

Maximum # Entity Expansions

Limit the number of entity expansions allowed to the specified number. Default: 1024.

Maximum Entity Expansion Depth

Restrict the maximum number of nested entity expansions to no more than the specified number. Default: 32

Maximum # Namespaces

Limit the number of namespace declarations in an XML document to no more than the specified number. Default: 16

Maximum Namespace URI Length

Limit the URL length of each namespace declaration to no more than the specified number of characters. Default: 256

Block Processing Instructions

Block any special processing instructions included in the request. This rule has no user-modifiable values.

Block DTD

Block any document type definitions (*DTD*) included with the request. This rule has no user-modifiable values.

Block External Entities

Block all references to external entities in the request. This rule has no user-modifiable values.

SOAP Array Check

Enable or disable the following SOAP array checks:

- **Maximum SOAP Array Size.** The maximum total size of all SOAP arrays in an XML request before the connection is blocked. You can modify this value. Default: 20000000
- **Maximum SOAP Array Rank.** The maximum rank or dimensions of any single SOAP array in an XML request before the connection is blocked. You can modify this value. Default: 16

XML Cross-Site Scripting Check

The XML Cross-Site Scripting check examines both the headers and the bodies of user requests for possible cross-site scripting attacks. If it finds a possible cross-site scripting attack, it blocks the request.

To prevent misuse of the scripts on your protected web services to breach security on your web services, the XML Cross-Site Scripting check blocks scripts that violate the *same origin rule*, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of Javascript-enhanced web content that violates the same origin rule. If you enable the XML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity. In addition, to prevent blocking of legitimate requests, this check ignores cookies that were set by the server, even if they contain elements that the Cookie Consistency check would otherwise block. You should keep that in mind when configuring this check.

If you use the wizard or the configuration utility, in the **Modify Cookie Consistency Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions, and the following parameters:

- **Transform.** Make the following changes to requests that match the XML Cross-Site Scripting check:

  - Left angle bracket (<) to HTML character entity equivalent (<)

  - Right angle bracket (>) to HTML character entity equivalent (>)

  These changes prevent browsers from interpreting unsafe html tags, such as `<script>`, and thereby executing malicious code. If you enable both request-header checking and transformation, any special characters found in request headers are also modified as described above. If scripts on your protected web site contain cross-site scripting features, but your web site does not rely upon those scripts to operate correctly, you can safely disable blocking and enable transformation. This configuration ensures that no legitimate web traffic is blocked, while stopping any potential cross-site scripting attacks.

  Check complete URLs. If checking of complete URLs is enabled, the application firewall examines entire URLs for XML Cross-Site Scripting attacks instead of checking just the query portions of URLs.

  Check Request headers. If Request header checking is enabled, the application firewall examines request headers for XML Cross-Site Scripting attacks, instead of examining just URLs.

If you use the command-line interface, you can enter the following commands to configure the XML Cross-Site Scripting Check:

- set appfw profile <profileName> **-crossSiteScriptingAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <profileName> **-crossSiteScriptingTransformUnsafeHTML** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-crossSiteScriptingCheckCompleteURLs** ([**ON**] | [**OFF**])

To specify relaxations for the XML Cross-Site Scripting check, you must use the configuration utility. On the **Checks** tab of the **Modify XML Cross-Site Scripting Check** dialog box, click **Add** to open the **Add XML Cross-Site Scripting Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify XML Cross-Site Scripting Check Relaxation**dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the Configuration Utility.

Following are examples of XML Cross-Site Scripting check relaxations:

Name element or attribute. The following expression exempts all elements beginning with the string `name_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^name_[0-9A-Za-z]{2,15}$
```

URL element or attribute. The following expression exempts URLs with hostnames of `web.example.com`, with a path up to four levels deep followed by an optional filename and extension, but no HTML or query symbols :

```
^https?://web[.]example[.]com(/[^<>?]{1,30}){0,4}(/[^<>?]{1,30})*$
```

URL element or attribute (special characters). The following expression exempts URLs with hostnames of `web.türkçe-example.com`, with the same path and file restrictions as above:

```
^https?://web[.]t\xC3\xBCrk\xC3\xA7e-example[.]com(/[^<>?]{1,30}){0,4}(/[^<>?]{1,30})*$
```

Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML Cross-Site Scripting check would otherwise have blocked.

XML SQL Injection Check

The XML SQL Injection check examines both the headers and the bodies of user requests for possible XML SQL Injection attacks. If it finds injected SQL, it blocks the request.

To prevent misusing the scripts on your protected web services to breach security on your web services, the XML SQL Injection check blocks scripts that violate the *same origin rule*, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called XML SQL Injection. The reason XML SQL Injection is a security issue is that a web server that allows XML SQL Injection can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of Javascript-enhanced web content that violates the same origin rule. If you enable the XML SQL Injection check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity. In addition, to prevent blocking of legitimate requests, this check ignores cookies that were set by the server, even if they contain elements that the Cookie Consistency check would otherwise block. You should keep this in mind when configuring this check.

Note: To prevent blocking of legitimate requests, this check ignores cookies that were set by the server, even if they contain elements that the Cookie Consistency check would otherwise block.

If you use the wizard or the configuration utility, in the **Modify XML SQL Injection Check** dialog box, on the **General** tab you can enable or disable **Block**, **Log**, and **Statistics** actions, and the following parameters:

- **Restrict checks to fields containing SQL special characters.** If you configure the application firewall to check only fields that contain SQL special strings, the application firewall skips web form fields that do not contain special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this parameter can significantly reduce the load on the application firewall and speed up processing without placing your protected web sites at risk.

- **SQL comments handling.** By default, the application firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if it is preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure the application firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:

  - **ANSI.** Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.

  - **Nested.** Skip nested SQL comments, which are normally used by Microsoft SQL Server.

  - **ANSI/Nested.** Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested

standard, are checked for injected SQL.

Caution: In most cases, you should not choose the Nested or the ANSI/Nested option unless your database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they may indicate an attempt to breach security on that server.

- **Check all Comments.** Check the entire request for injected SQL, without skipping anything. The default setting.

- **Check Request headers.** If Request header checking is enabled, the application firewall examines the headers of requests for XML SQL Injection attacks, instead of just URLs.

  Caution: If you enable both request header checking and transformation, any SQL special characters found in headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;), so enabling both Request header checking and transformation simultaneously may cause errors.

If you use the command-line interface, you can enter the following commands to configure the XML SQL Injection Check:

- set appfw profile <profileName> **-SQLInjectionAction** [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <profileName> **-SQLInjectionTransformSpecialChars** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-SQLInjectionOnlyCheckFieldsWithSQLChars** ([**ON**] | [**OFF**])
- set appfw profile <profileName> **-SQLInjectionParseComments** ([**checkall**] | [**ansi** | **nested**] | [**ansinested**])

You configure the exceptions to the XML SQL Injection check by opening the **Modify XML SQL Injection Check** dialog box, **Checks** tab. An exception can consist of either a literal string or a PCRE-format regular expression. For information about adding, modifying, removing, enabling, or disabling exceptions, see Manual Configuration By Using the Configuration Utility.

Following are examples of XML SQL Injection check relaxations:

- **Name element or attribute.** The following expression exempts all elements beginning with the string `name_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than fifteen characters long:

  ```
  ^name_[0-9A-Za-z]{2,15}$
  ```

- **URL element or attribute.** The following expression exempts URLs with hostnames of `web.example.com`, with a path up to four levels deep followed by an optional file name and extension, but no HTML or query symbols :

^https?://web[.]example[.]com(/[^<>?]{1,30}){0,4}(/[^<>?]{1,30})*$

- **URL element or attribute (special characters).** The following expression exempts URLs with hostnames of `web.türkçe-example.com`, with the same path and file restrictions as above:

  ^https?://web[.]t\xC3\xBCrk\xC3\xA7e-example[.]com(/[^<>?]{1,30}){0,4}(/[^<>?]{1,30})*$

  Note: See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

XML Attachment Check

The XML Attachment check examines incoming requests for malicious attachments, and it blocks those requests that contain attachments that might breach applications security. The purpose of the XML Attachment check is to prevent an attacker from using an XML attachment to breach security on your server.

If you use the wizard or the configuration utility, in the **Modify XML Attachment Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, **Statistics**, and **Learn** actions:

If you use the command-line interface, you can enter the following command to configure the XML Attachment Check:

- set appfw profile <profileName> **-xmlAttachmentAction [block] [log] [stats] [none]**

You must configure the other XML Attachment check settings in the configuration utility. In the **Modify XML Attachment Check** dialog box, on the **Checks** tab, you can configure the following settings:

- **Maximum Attachment Size.** Allow attachments that are no larger than the maximum attachment size you specify. To enable this option, first select the **Enabled** check box, and then type the maximum attachment size in bytes in the **Size** text box.
- **Attachment Content Type.** Allow attachments of the specified content type. To enable this option, first select the **Enabled** check box, and then enter a regular expression that matches the Content-Type attribute of the attachments that you want to allow.
    - You can type the URL expression directly in the text window. If you do so, you can use the **Regex Tokens** menu to enter a number of useful regular expressions at the cursor instead of typing them manually.
    - You can click **Regex Editor** to open the **Add Regular Expression** dialog box and use it to construct the URL expression.

Web Services Interoperability Check

The Web Services Interoperability (WS-I) check examines both requests and responses for adherence to the WS-I standard, and blocks those requests and responses that do not adhere to this standard. The purpose of the WS-I check is to block requests that might not interact with other XML appropriately. An attacker can use inconsistencies in interoperability to launch an attack on your XML application.

If you use the wizard or the configuration utility, in the **Modify Web Services Interoperability Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, **Statistics**, and **Learn** actions.

If you use the command-line interface, you can enter the following command to configure the Web Services Interoperability check:

- set appfw profile <profileName> **-xmlWSIAction** [**block**] [**log**] [**learn**] [**stats**] [**none**]

To configure individual Web Services Interoperability rules, you must use the configuration utility. On the **Checks** tab of the **Modify Web Services Interoperability Check** dialog box, select a rule and click **Enable** or **Disable** to enable or disable the rule. You can also click **Open** to open the **Web Services Interoperability Detail** message box for that rule. The message box displays read-only information about the rule. You cannot modify or make other configuration changes to any of these rules.

XML Message Validation Check

The XML Message Validation check examines requests that contain XML messages to ensure that they are valid. If a request contains an invalid XML message, the application firewall blocks the request. The purpose of the XML Validation check is to prevent an attacker from using specially constructed invalid XML messages to breach the security of your application.

If you use the wizard or the configuration utility, in the **Modify XML Message Validation Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions.

If you use the command-line interface, you can enter the following command to configure the XML Message Validation Check:

- set appfw profile <profileName> **-xmlValidationAction [block] [log] [stats] [none]**

You must use the configuration utility to configure the other XML Validation check settings. In the **Modify XML Message Validation Check** dialog box, on the **Checks** tab, you can configure the following settings:

- **XML Message Validation.** Use one of the following options to validate the XML message:
    - **SOAP Envelope.** Validate only the SOAP envelope of XML messages.
    - **WSDL.** Validate XML messages by using an XML SOAP WSDL. If you choose WSDL validation, in the **WSDL Object** drop-down list you must choose a WSDL. If you want to validate against a WSDL that has not already been imported to the application firewall, you can click the **Import** button to open the **Manage WSDL Imports** dialog box and import your WSDL. See WSDL for more information.
        - If you want to validate the entire URL, leave the **Absolute** radio button in the **End Point Check** button array selected. If you want to validate only the portion of the URL after the host, select the **Relative** radio button.
        - If you want the application firewall to enforce the WSDL strictly, and not allow any additional XML headers not defined in the WSDL, you must clear the **Allow additional headers not defined in the WSDL** check box.

          Caution: If you uncheck the **Allow Additional Headers not defined in the WSDL** check box, and your WSDL does not define all XML headers that your protected XML application or Web 2.0 application expects or that a client sends, you may block legitimate access to your protected service.
    - **XML Schema.** Validate XML messages by using an XML schema. If you choose XML schema validation, in the **XML Schema Object** drop-down list you must choose an XML schema. If you want to validate against an XML schema that has not already been imported to the application firewall, you can click the **Import** button to open the **Manage XML Schema Imports** dialog box and import your WSDL. See WSDL for more information.
- **Response Validation.** By default, the application firewall does not attempt to validate responses. If you want to validate responses from your protected application or Web 2.0 site, select the **Validate Response** check box. When you do, the **Reuse the XML**

Schema specified in request validation check box and the **XML Schema Object** drop-down list are activated.

- Check the **Reuse XML Schema** check box to use the schema you specified for request validation to do response validation as well.

  Note: If you check this check box, the **XML Schema Object** drop-down list is grayed out.

- If you want to use a different XML schema for response validation, use the **XML Schema Object** drop-down list to select or upload that XML schema .

XML SOAP Fault Filtering Check

The XML SOAP Fault Filtering check examines responses from your protected web services and filters out XML SOAP faults. This prevents leaking of sensitive information to attackers.

If you use the wizard or the configuration utility, in the **Modify XML SOAP Fault Filtering Check** dialog box, on the **General** tab you can enable or disable the **Block**, **Log**, and **Statistics** actions, and the **Remove** action, which removes SOAP faults before forwarding the response to the user.

If you use the command-line interface, you can enter the following command to configure the XML SOAP Fault Filtering Check:

set appfw profile <profileName> **-XMLSOAPFaultAction** [**block**] [**log**] [**stats**] [**none**]

You cannot configure exceptions to the XML SOAP Fault Filtering check. You can only enable or disable it.

Policies

The application firewall uses two types of policies: firewall policies and auditing policies. Firewall policies control which traffic is sent to the application firewall. Auditing policies control the log server to which application firewall logs are sent.

Firewall policies can be complex because the policy rule can consist of multiple expressions in the NetScaler expressions language, which is a full-fledged object oriented programming language capable of defining with extreme precision exactly which connections to filter. Because firewall policies operate within the context of the application firewall, they must meet certain criteria that are connected to how the application firewall functions and what traffic is appropriately filtered by it. As long as you keep these criteria in mind, however, firewall policies are similar to policies for other NetScaler features. The instructions here do not attempt to cover all aspects of writing firewall policies, but only provide an introduction to policies and cover those criteria that are unique to the application firewall.

Auditing policies are simple because the policy rule is always `ns_true`. You need only specify the log server that you want to send logs to, the logging levels that you want to use, and a few other criteria that are explained in detail.

Firewall Policies

A firewall policy is a rule associated with a profile. The rule is an expression or group of expressions that defines the types of request/response pairs that the application firewall is to filter by applying the profile. Firewall policy expressions are written in the NetScaler expressions language, an object-oriented programming language with special features to support specific NetScaler functions. The profile is the set of actions that the application firewall is to use to filter request/response pairs that match the rule.

Firewall policies enable you to assign different filtering rules to different types of web content. Not all web content is alike. A simple web site that uses no complex scripting and accesses and handles no private data might require only the level of protection provided by a profile created with basic defaults. Web content that contains Javascript-enhanced web forms or accesses an SQL database probably requires more tailored protection. You can create a different profile to filter that content, and create a separate firewall policy that can determine which requests are attempting to access that content. You then associate the policy expression with a profile you created and globally bind the policy to put it into effect.

The application firewall processes only HTTP connections, and therefore uses a subset of the overall NetScaler expressions language. The information here is limited to topics and examples that are likely to be useful when configuring the application firewall. For a procedure that explains how to create and configure a policy by using the configuration utility, see To create and configure a policy. For a procedure to do so by using the NetScaler command line, see To create and configure a policy. For detailed information about the NetScaler expressions language, see the *Citrix NetScaler Policy Configuration and Reference Guide*.

To create an Application Firewall rule (expression) by using the configuration utility

Like other NetScaler policy rules, application firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple policy, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility to create your policy rule:

    - If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click **Application Firewall**, then in the details pane click **Application Firewall Wizard**, and then navigate to the **Specify Rule** screen.

    - If you are configuring a policy manually, in the navigation pane, expand **Application Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a new policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.

2. On the **Specify Rule** screen, the **Create Application Firewall Profile** dialog box, or the **Configure Application Firewall Profile** dialog box, click **Prefix**, and then choose the prefix for your expression from the drop-down list. Your choices are:

   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

   After you choose a prefix, the application firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.

3. Choose your next term.

   If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The application firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

   When you have decided which term you want, double-click it to insert it into the **Expression** window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose `HTTP.REQ.HEADER("")`, type the header name between the quotation marks.

5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

   Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

     HTTP.REQ.HEADER("Host").EQ("shopping.example.com")

     For `shopping.example.com`, substitute the name of the web host that you want to match.

   - **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

     HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")

For `www.example.com`, substitute the name of the web host. For `folder`, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called `/solutions/orders`, you substitute that string for folder.

- **Specific type of content: GIF images.** To match GIF format images:

  HTTP.REQ.URL.ENDSWITH(".gif")

  To match other format images, substitute another string in place of `.gif`.

- **Specific type of content: scripts.** To match all CGI scripts located in the `CGI-BIN` directory:

  HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")

  To match all JavaScripts with `.js` extensions:

  HTTP.REQ.URL.ENDSWITH(".js")

  For more information about creating policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide*.

The Add Expression Dialog Box

The **Add Expression** dialog box (also referred to as the *Expression Editor*) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility:
   - If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click **Application Firewall**, then in the details pane click **Application Firewall Wizard**, and then navigate to the **Specify Rule** screen.
   - If you are configuring a policy manually, in the navigation pane, expand **Application Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a new policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, in the **Create Application Firewall Profile** dialog box, or in the **Configure Application Firewall Profile** dialog box, click **Add**.
3. In the **Add Expression** dialog box, in the **Construct Expression** area, in the first list box, choose one of the following prefixes:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.

- **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

- **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

- **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the application firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The **Help** window displays a description of the second term, and the **Preview Expression** window displays your expression.

5. In the third list box, choose the next term. A new list box appears to the right, and the **Help** window changes to display a description of the new term. The **Preview Expression** window updates to display the expression as you have specified it to that point.

6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.

7. When you have finished constructing your expression, click **OK** to close the **Add Expression** dialog box. Your expression is inserted into the **Expression** text area.

Auditing Policies

Auditing policies determine the messages that are generated and logged during an Application Firewall session. These messages are logged in SYSLOG format to the local NSLOG server or to an external logging server. Different types of messages are logged on the basis of the level of logging selected.

To create an auditing policy, you must first create either an NSLOG server or a SYSLOG server. After specifying the server, you create the policy and specify the type of log and the server to which logs are sent.

To create an auditing server by using the NetScaler command line

You can create two different types of auditing server: an NSLOG server or a SYSLOG server. The command names are different, but the parameters for the commands are the same.

To create an auditing server, at the NetScaler command prompt, type the following commands:

- add audit <type> <name> <serverIP> [**-serverPort** <port>] **-logLevel** <logLevel> ... [**-dateFormat** (**MMDDYYYY** | **DDMMYYYY**)] [**-logFacility** <logFacility>] [**-tcp** (**NONE** | **ALL**)] [**-acl** (**ENABLED** | **DISABLED**)] [**-timeZone** (**GMT_TIME** | **LOCAL_TIME**)] [**-userDefinedAuditlog** (**YES** | **NO**)] [**-appflowExport** (**ENABLED** | **DISABLED**)]

- save ns config

Example

The following example creates a syslog server named syslog1 at IP 10.124.67.91, with loglevels of emergency, critical, and warning, log facility set to LOCAL1, that logs all TCP connections:

```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning -logFacility LOCAL1 -tcp ALL
save ns config
```

To modify or remove an auditing server by using the NetScaler command line

- To modify an auditing server, type the set audit <type> command, the name of the auditing server, and the parameters to be changed, with their new values.
- To remove an auditing server, type the rm audit <type> command and the name of the auditing server.

Example

The following example modifies the syslog server named syslog1 to add errors and alerts to the log level:

```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning alert error
-logFacility LOCAL1 -tcp ALL
save ns config
```

Parameters for configuring an auditing server

type (Auditing Type)

The type of logging server you are configuring. Possible values: nslogAction (NSLOG), syslogAction (SYSLOG).

name (Name)

A name for the syslog server. The name can consist of from one to 31 upper-case and lower-case letters, numbers, and the period (.), underscore (_) and hyphen (-) symbols.

serverIP (IP Address)

The IP address of the server, in IPV4 or IPV6 format.

serverPort (Port)

The port number on which the server listens for connections.

logLevel (Log Levels)

The types of information to be logged to the server. The choices are:

- ALERT
- CRITICAL
- DEBUG

- EMERGENCY
- ERROR
- INFORMATIONAL
- NOTICE
- WARNING

dateFormat (Date format)

The format used for dates in the logs. The choices are MMDDYYYY (U.S. style) or DDMMYYYY (International style).

logFacility (Log Facility)

The log facility on the NetScaler appliance or VPX virtual appliance. Possible values: LOCAL0, LOCAL1, LOCAL2.

tcp (TCP Logging)

Enable logging of TCP connections. Possible values: NONE, ALL.

acl (ACL Logging)

Enable logging of ACL connections. Possible values: ENABLED, DISABLED.

timeZone (Time Zone)

The Unix time zone to use in the logs.

userDefinedAuditLog (User Configurable Log Messages)

Enable user configurable log messages. Possible values: YES, NO.

appflowExport (AppFlow Logging)

Enable export of logs to the NetScaler AppFlow feature. Possible values: ENABLED, DISABLED.

Example

The following example creates a syslog server named syslog1 at IP 10.124.67.91, with loglevels of emergency, critical, and warning, log facility set to LOCAL1, that logs all TCP connections:

```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning
-logFacility LOCAL1 -tcp ALL
save ns config
```

To create or configure an auditing server by using the configuration utility

1. In the navigation pane, expand **Application Firewall**, then **Policies**, and then **Auditing**.
2. In the details pane, click the **Server** tab.
3. Do one of the following:
   - To add a new auditing server, click **Add**.
   - To modify an existing auditing server, select the server, and then click **Open**.
4. In the **Create Auditing Server** or **Configure Auditing Server** dialog box, set the following parameters:
   - Name
   - Auditing Type
   - IP Address
   - Port
   - Log Levels
   - Log Facility
   - TCP Logging
   - ACL Logging
   - User-Configurable Log Messages
   - AppFlow Logging
   - Date Format
   - Time Zone
5. Click **Create** or **OK**.

To create an auditing policy by using the NetScaler command line

You can create an NSLOG policy or a SYSLOG policy. The type of policy must match the type of server. The command names for the two types of policy are different, but the parameters for the commands are the same.

At the NetScaler command prompt, type the following commands:

- add audit <type> <name> [**-rule** <expression>] [**-action** <string>]

- save ns config

Example

The following example creates a policy named syslogP1 that logs application firewall traffic to a syslog server named syslog1.

```
add audit syslogPolicy syslogP1 -rule "ns_true" -action syslog1
save ns config
```

To configure an auditing policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- set audit <type> <name> [**-rule** <expression>] [**-action** <string>]
- save ns config

Example

The following example modifies the policy named syslogP1 to log application firewall traffic to a syslog server named syslog2.

```
set audit syslogPolicy syslogP1 -rule "ns_true" -action syslog2
save ns config
```

Parameters for an auditing policy

type (Auditing Type)

The type of syslog server that you are using. The choices are nslogPolicy (NSLOG) and syslogPolicy (SYSLOG).

name (Name)

A name for the syslog server. The name can consist of from one to 31 upper-case and lower-case letters, numbers, and the period (.), underscore (_) and hyphen (-) symbols.

rule (no configuration utility equivalent)

The rule that defines the policy. Always `ns_true` for application firewall policies. Must be included when configuring an auditing policy at the NetScaler command line. Is included automatically when configuring by using the configuration utility.

action (Server)

The name of the auditing server.

To configure an auditing policy by using the configuration utility

1. In the navigation pane, expand **Application Firewall**, then **Policies**, and then **Auditing**.
2. In the details pane, do one of the following:
   - To add a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Open**.
3. In the **Create Auditing Policy** or **Configure Auditing Policy** dialog box, set the following parameters:
   - Name
   - Auditing Type
   - Server
4. Click **Create** or **OK**.

Imports

Several application firewall features make use of external files that you upload to the application firewall when configuring it. Using the configuration utility, you manage those files in the **Imports** pane, which has four tabs corresponding to the four types of files you can import: HTML error objects, XML error objects, XML schemas, and Web Services Description Language (WSDL) files. Using the NetScaler command line, you can import these types of files, but you cannot export them.

HTML Error Object

When a user's connection to an HTML or Web 2.0 page is blocked, or a user asks for a non-existent HTML or Web 2.0 page, the application firewall sends an HTML-based error response to the user's browser. When configuring which error response the application firewall should use, you have two choices:

- You can configure a *redirect URL*, which can be hosted on any Web server to which users also have access. For example, if you have a custom error page on your Web server, `404.html`, you can configure the application firewall to redirect users to that page when a connection is blocked.

- You can configure an *HTML error object*, which is an HTML-based Web page that is hosted on the application firewall itself. If you choose this option, you must upload the HTML error object to the application firewall. You do that in the Imports pane, on the **HTML Error Object** tab.

The error object must be a standard HTML file that contains no non-HTML syntax except for application firewall error object customization variables. It cannot contain any CGI scripts, server-parsed code, or PHP code. The customization variables enable you to embed troubleshooting information in the error object that the user receives when a request is blocked. While most requests that the application firewall blocks are illegitimate, even a properly configured application firewall can occasionally block legitimate requests, especially when you first deploy it or after you make significant changes to your protected Web sites. By embedding information in the error page, you provide the user with the information that he or she needs to give to the technical support person so that any issues can be fixed.

The application firewall error page customization variables are:

- `${NS_TRANSACTION_ID}`. The transaction ID that the application firewall assigned to this transaction.

- `${NS_APPFW_SESSION_ID}`. The application firewall session ID.

- `${NS_APPFW_VIOLATION_CATEGORY}`. The specific application firewall security check or rule that was violated.

- `${NS_APPFW_VIOLATION_LOG}`. The detailed error message associated with the violation.

- `${COOKIE("<CookieName>")}`. The contents of the specified cookie. For <CookieName>, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.

  Note: If you have blocking enabled for the Cookie Consistency Check, any blocked cookies are not displayed on the error page because the application firewall blocks them.

To use these variables, you embed them in the HTML or XML of the error page object as if they were an ordinary text string. When the error object is displayed to the user, for each customization variable the application firewall substitutes the information to which the variable refers. An example HTML error page that uses custom variables is shown below.

```
<!doctype html public "-//w3c//dtd html 4.0//en">
<html>
<head>
<title>Page Not Accessible</title>
</head>
<body>
<h1>Page Not Accessible</h1>
<p>The page that you accessed is not available. You can:</p>
<ul>
<li>return to the <b><a href="[homePage]">home page</a></b>, re-establish your session, and try again, or,
<li>report this incident to the help desk via <b><a href="mailto:[helpDeskEmailAddress]">email</a></b> or b
</ul>
<p>If you contact the help desk, please provide the following information:</p>
<table cellpadding=8 width=80%>
<tr><th align="right" width=30%>Transaction ID:</th><td align="left" valign="top" width=70%>${NS_TRANSACT
<tr><th align="right" width=30%>Session ID:</th><td align="left" valign="top" width=70%>${NS_APPFW_SESSIO
<tr><th align="right" width=30%>Violation Category:</th><td align="left" valign="top" width=70%>${NS_APPFW
<tr><th align="right" width=30%>Violation Log:</th><td align="left" valign="top" width=70%>${NS_APPFW_VIO
<tr><th align="right" width=30%>Cookie Name:</th><td align="left" valign="top" width=70%>${COOKIE("[cooki
</table>
<body>
<html>
```

To use this error page, copy it into a text or HTML editor. Substitute the appropriate local information for the following variables, which are enclosed in square brackets to distinguish them from the NetScaler variables. (Leave those unchanged.):

- **[homePage]**. The URL for your web site's home page.
- **[helpDeskEmailAddress]**. The email address that you want users to use to report blocking incidents.
- **[helpDeskPhoneNumber]**. The phone number that you want users to call to report blocking incidents.
- **[cookieName]**. The name of the cookie whose contents you want to display on the error page.

XML Error Object

When a user's connection to an XML page is blocked, or a user asks for a nonexistent XML application, the application firewall sends an XML-based error response to the user's browser. You configure the error response by uploading an XML-based error page to the application firewall in the **Imports Pane**, on the **XML Error Object** tab. All XML error responses are hosted on the application firewall. You cannot configure a redirect URL for XML applications.

Note: You can use the same customization variables in an XML error object as in an HTML error object.

XML Schema

When the application firewall performs a validation check on a user's request for an XML or Web 2.0 application, it can validate the request against the XML schema or design type document (DTD) for that application and reject any request that does not follow the schema or DTD. Both an XML schema and a DTD are standard XML configuration files that describe the structure of a specific type of XML document.

WSDL

When the application firewall performs a validation check on a user's request for an XML SOAP-based web service, it can validate the request against the web services type definition (*WSDL*) file for that web service. A WSDL file is a standard XML SOAP configuration file that defines the elements of a specific XML SOAP web service.

Importing and Exporting Files

You can import HTML or XML error objects, XML schemas, DTDs, and WSDLs to the application firewall by using the configuration utility or the NetScaler command line. To export any of these files or objects, or to edit a file or object directly on the application firewall, you must use the configuration utility.

To import a file or object by using the NetScaler command line

At the command prompt, type the following commands:

- import appfw <type> <src> <name>
- save ns config

Example

The following example imports an HTML error object from a file named `error.html` and assigns it the name `HTMLError`.

```
import htmlerrorpage error.html HTMLError
save ns config
```

Parameters for Importing a File or Object

type (Type)

The type of file or object that you are uploading. Possible Values: `HTMLErrorPage`, `xmlerrorpage`, `xmlschema`, `wsdl`.

src (Source File)

The full URL or path and filename of the object or file that you want to upload.

- If you are uploading the object from a web site or intranet location, you should type a URL in standard browser format.
-

If you are importing the object from your local computer, you should type the path and file name of the object in the appropriate format for your local computer, or use the browse dialog to locate the file.

name (Name)

The name to be assigned to the file or object on the application firewall. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

To import a file or object by using the configuration utility

Before you attempt to import an XML schema, DTD, or WSDL file, or an HTML or XML error object from a network location, verify that the NetScaler appliance can connect to the Internet or LAN computer where the file is located. Otherwise, you cannot import the file or object.

1. In the navigation pane, expand **Application Firewall**, and then select **Imports**.
2. In the **Application Firewall Imports** pane, select the tab for the type of file you want to import, and then click **Add**.

   The upload process is identical on all four tabs from the user point of view.
3. In the **Import New** dialog box, **Name** text box, type a name for the object you are importing.
4. Choose the type of upload.

   - If the object is on a web site or other location on a LAN, WAN, or the Internet, select **Import from URL**.
   - f the object is on your local computer or a file server mounted on your local computer, select **Import from Local File**.
5. In the **URL** or **Local File** text box, type the full URL or path and filename to the resource.
6. Click **Import**.

   - If the application firewall finds the specified resource, the Import Console message box notifies you that the import succeeded. Click **Close**.
   - If the application firewall is unable to locate the resource, and you are uploading from an Internet or Intranet site, and you have verified that the URL and file you requested exists, click **Close** to close the **Import Console** message box. Next, check the NetScaler log file (`ns.log`) to verify that the URL or path and file that you used is accessible from your application firewall. If it is not, fix the access issue or move the file or object to a location that is accessible, and then repeat steps 4 and 5 to import the object.

7. To delete an object, select the object, and then click **Remove**. When the **Proceed** dialog box appears, click **OK**.

To export a file or object

Before you attempt to export an XML schema, DTD, or WSDL file, or an HTML or XML error object from a network location, verify that the application firewall appliance can connect to the Internet or LAN computer where the file is located. Otherwise, you cannot export the file or object.

1. In the configuration utility's navigation pane, expand **Application Firewall**, and then select **Imports**.
2. In the **Application Firewall Imports** pane, select the tab for the type of file you want to export.

   The export process is identical on all four tabs from the user point of view.
3. Select the file that you want to export, and then click **Export**.
4. In the **Export** dialog box, click **Browse**, navigate to a local file system and directory in which to save the file or object that you are exporting, and click **Select**.
5. Click **Export**.

To edit a file or object

In the configuration utility, you can modify the properties of configuration files that you previously uploaded to the application firewall. You can also edit the text of HTML and XML error objects directly in the configuration utility.

1. In the navigation pane, expand **Application Firewall**, then select **Imports**, and then select the tab for the type of file that you want to modify.
2. Select the file that you want to modify, and then click **Open**.

   If the object is an HTML or XML error object, the text of the object is displayed in a window. You can modify the text by using the standard browser-based editing tools and methods for your browser.

   Note: The edit window is designed to allow you to make minor changes to your HTML or XML error object. To make extensive changes, you may prefer to export the error object to your local computer and use standard HTML or XML web page editing tools.

   If the object is an XML schema, DTD, or WSDL file, the name and URL of the object are displayed in a dialog box. The name is read-only. You can modify the URL.
3. Click **OK** (for HTML or XML error objects) or **Import** (for XML schemas, DTDs, or WSDLs) to save your changes, and then click **Close**.

Global Configuration

The application firewall global configuration affects all profiles and policies. The Global Configuration items are:

- **Engine Settings.** A collection of global settings—session cookie name, session time-out, maximum session lifetime, logging header name, undefined profile, default profile, and import size limit—that pertain to all connections that the application firewall processes, rather than to a specific subset of connections.

- **Confidential Fields.** A set of form fields in web forms that contain sensitive information that should not be logged to the application firewall logs. Form fields such as password fields on a logon page or credit card information on a shopping cart checkout form are normally designated as confidential fields.

- **Field Types.** The list of web form field types used by the Field Formats security check. Each of these field types is defined by a PCRE-compliant regular expression that defines the type of data and the minimum/maximum length of data that should be allowed in that type of form field.

Engine Settings

The engine settings affect all connections that the application firewall processes. They include the name of the cookie that stores the session ID, the session time-out and maximum lifetime values, the client IP address logging header, the profile to apply when the corresponding policy action is undefined, the default profile (applied to connections that do not match a policy), and the size limit for imported files. Normally, the default values for these settings are correct. If the default settings cause a conflict with other servers or cause premature disconnection of your users, however, you may need to modify them.

To configure engine settings by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- set appfw settings [**-sessionCookieName** <name>] [**-sessionTimeout** <positiveInteger>] [**-cookiePostEncryptPrefix** <string>] [**-sessionLifetime** <positive integer>] [**-clientIPLoggingHeader** <headerName>] [**-undefaction** <profileName>] [**-defaultProfile** <profileName>] [**-importSizeLimit** <positiveInteger>]

- save ns config

Example

```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout 3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096
save ns config
```

Parameters for configuring engine settings

sessionCookieName (Cookie Name)

The name of the session cookie, which is a cookie that the application firewall uses to track user sessions. You do not normally need to modify the name of this cookie, but if it conflicts in any way with a cookie set by your protected web servers, you can change it. The cookie name must begin with a letter or number, and can consist of from 1 to 31 letters, numbers, and the hyphen (-) and underscore (_) symbols. Default: `citrix_ns_id`.

sessionTimeout (Session Timeout)

The length of time, in seconds, that the application firewall waits before timing out user sessions. After the application firewall times out the session, the user must reestablish a session by visiting the home page or a designated start URL. Possible values: 1 to 600 seconds. Default: 900 seconds (15 minutes).

cookiePostEncryptPrefix (Cookie Post Encrypt Prefix)

The string that is prepended to all encrypted cookie values. Default: ENC

sessionLifetime (Maximum Session Lifetime)

The maximum amount of time, in seconds, that the application firewall allows a user session to remain active, regardless of user activity. When the limit is reached, the application firewall terminates the session. To regain access, the user must establish a new session by visiting a designated start page. Possible values: Disabled, or any value from 1 through 14,400 seconds. Default: 900 seconds (15 minutes).

clientIPLoggingHeader (Logging Header Name)

The name of an HTTP header containing the IP address that the client used to connect to your protected web site or service. Possible values: Any value that the HTTP server supports. Default: Null value (do not add a logging header).

undefAction (Undefined Profile)

The profile to use when an application firewall policy evaluates as undefined. An undefined evaluation indicates an internal error condition. If such an error occurs when evaluating a classic policy, the application firewall aborts processing of the associated connections and passes them back to the NetScaler appliance without attempting to filter them. This behavior can constitute a security hazard in some circumstances, so the default setting specifies the APPFW_BLOCK built-in profile. You can specify a different built-in or user-created profile as the undefined profile.

defaultProfile (Default Profile)

The profile to use when a connection does not match any of the policies that you have defined on the application firewall. Normally the default profile is set to APPFW_BYPASS, which configures the application firewall to send connections that fail to match any policy back to the NetScaler appliance without attempting to filter them further.

importSizeLimit (Import Size Limit)

The cumulative total maximum number of bytes allowed for importing files to the application firewall. If the total size of uploaded files in a web form is larger than the configured limit, the application firewall blocks the request. Possible value: Any number. Default: 0 (disabled).

To configure engine settings by using the configuration utility

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, click **Change Engine Settings**.
3. In the **Application Firewall Engine Settings** dialog box, set the following parameters:
    - Cookie Name
    - Session Timeout
    - Cookie Post Encrypt Prefix
    - Maximum Session Lifetime
    - Logging Header Name
    - Undefined Profile
    - Default Profile
    - Import Size Limit
4. Click **OK**.

Confidential Fields

You can designate web-form fields as confidential to protect the information users type into them. Normally, any information a user types into a web form on one of your protected web servers is logged in the NetScaler logs. The information typed into a web-form field designated as confidential, however, is not logged. That information is saved only where the web site is configured to save such data, normally in a secure database.

Common types of information that you may want to protect with a confidential field designation include:

- Passwords
- Credit card numbers, validation codes, and expiration dates
- Social security numbers
- Tax ID numbers
- Home addresses
- Private telephone numbers

In addition to being good practice, proper use of confidential field designations may be necessary for PCI-DSS compliance on ecommerce servers, HIPAA compliance on servers that manage medical information in the United States, and compliance with other data protection standards.

Important: In the following two cases, the Confidential Field designation does not function as expected:

- If a Web form has either a confidential field or an action URL longer than 256 characters, the field or action URL is truncated in the NetScaler logs.
- With certain SSL transactions, the logs are truncated if either the confidential field or the action URL is longer than 127 characters.

In either of these cases, the application firewall masks a fifteen-character string with the letter "x," instead of the normal eight character string. To ensure that any confidential information is removed, the user must use form field name and action URL expressions that match the first 256, or (in cases where SSL is used) the first 127 characters.

To configure your application firewall to treat a web-form field on a protected web site as confidential, you add that field to the Confidential Fields list. You can enter the field name as a string, or you can enter a PCRE-compatible regular expression specifying one or more fields. You can enable the confidential-field designation when you add the field, or you can modify the designation later.

To add a confidential field by using the NetScaler command line

At the command prompt, type the following commands:

- add appfw confidField <name> "<url>" [**-isRegex** (REGEX | NOTREGEX)] [**-comment** "<string>"] [**-state** (ENABLED | DISABLED)]
- save ns config

Example

The following example adds all web form fields whose names begin with`Password` to the confidential fields list.

```
add appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cg
save ns config
```

To modify a confidential field by using the NetScaler command line

At the command prompt, type the following commands:

- set appfw confidField <name> "<url>" [**-isRegex** (REGEX | NOTREGEX)] [**-comment** "<string>"] [**-state** (ENABLED | DISABLED)]
- save ns config

Example

The following example modifies the confidential field designation to add a comment.

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi
save ns config
```

To remove a confidential field by using the NetScaler command line

At the command prompt, type the following commands:

- rm appfw confidField <name> "<url>"
- save ns config

Parameters for configuring a confidential field

state (Enable)

Enable the field's designation as a confidential field.

name (Field Name)

The name of the web form field that you are designating as confidential.

isRegex (Is form field name a regular expression)

Is the string that you used to define the form field namea regular expression or not?

url (Action URL)

The URL of the web page that contains that web form.

comment (Comments)

A comment. Optional.

To configure a confidential field by using the configuration utility

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, under **Settings**, click **Manage Confidential Fields**.
3. In the Manage Confidential Fields dialog box, do one of the following:
   - To add a new form field to the list, click **Add**.
   - To change an existing confidential field designation, select the field, and then click **Open**.

   The **Create Confidential Form Field** dialog box or the **Configure Confidential Form Field** dialog box appears.

   Note: If you select an existing confidential field designation and then click **Add**, the **Create Confidential Form Field** dialog box displays the information for that confidential field. You can modify that information to create your new confidential field.
4. In the dialog box, fill out the elements. They are:
   - **Enabled check box.** Select or clear to enable/disable this confidential field designation.
   - **Is form field name a regular expression check box.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **Field Name.** Enter a literal string or PCRE-format regular expression that either represents a specific field name or that matches multiple fields with names that follow a pattern.
   - **Action URL.** Enter a literal URL or a regular expression that defines one or more URLs of the web page(s) on which the web form(s) that contains the confidential field are located.
   - **Comments.** Enter a comment. Optional.
5. Click **Create** or **OK**.
6. To remove a confidential field designation from the confidential fields list, select the confidential field listing you want to remove, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding, modifying, and removing confidential field designations, click **Close**.

Examples

Following are some regular expressions that define form field names that you might find useful:

- `^passwd_` (Applies confidential-field status to all field names that begin with the "passwd_" string.)

- `^(([0-9a-zA-Z._-]*||\\x[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_` (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)

Following are some regular expressions that define specific URL types that you might find useful. Substitute your own web host(s) and domain(s) for those in the examples.

- If the web form appears on multiple web pages on the web host `www.example.com`, but all of those we named `logon.pl?`, you could use the following regular expression:

  https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon[.]pl\?

- If the web form appears on multiple web pages on the web host `www.example-español.com`, which c n-tilde (ñ) special character, you could use the following regular expression, which represents the n-tild character as an encoded UTF-8 string containing C3 B1, the hexadecimal code assigned to that character charset:

  https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)* logon[.]pl\?

- If the web form containing `query.pl` appears on multiple web pages on different hosts within the exa domain, you could use the following regular expression:

  https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?

- If the web form containing `query.pl` appears on multiple web pages on different hosts in different dor could use the following regular expression:

  https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{2,6}/([0-9A-Za-z][0-9A-Za-z_-.

- If the web form appears on multiple web pages on the web host `www.example.com`, but all of those we named `logon.pl?`, you could use the following regular expression:

  https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?

Field Types

A field type is a PCRE-format regular expression that defines a particular data format and minimum/maximum data lengths for a form field in a web form. Field types are used in the Field Formats check.

The application firewall comes with several default field types, which are:

- `integer`. A string of any length consisting of numbers only, without a decimal point, and with an optional preceding minus sign (-).
- `alpha`. A string of any length consisting of letters only.
- `alphanum`. A string of any length consisting of letters and/or numbers.
- `nohtml`. A string of any length consisting of characters, including punctuation and spaces, that does not contain HTML symbols or queries.
- `any`. Anything at all.

  Important: Assigning the `any` field type as the default field type, or to a field, allows active scripts, SQL commands, and other possibly dangerous content to be sent to your protected web sites and applications in that form field. You should use the any type sparingly, if you use it at all.

You can also add your own field types to the Field Types list. For example, you might want to add a field type for a social security number, postal code, or phone number in your country. You might also want to add a field type for a customer identification number or store credit card number.

To add a field type to the Field Types list, you enter the field name as a literal string or PCRE-format regular expression.

To add a field type by using the NetScaler command line

At the command prompt, type the following commands:

- add appfw fieldType <name> "<rule>" <priority> [**-comment** "<string>"]
- save ns config

Example

The following example adds a field type named SSN that matches US Social Security numbers to the Field Types list, and sets its priority to 1.

```
add appfw fieldType SSN "^[1-9][0-9]{2,2}-[0-9}{2,2}-[0-9]{4,4}$" 1
save ns config
```

To modify a field type by using the NetScaler command line

At the command prompt, type the following commands:

- set appfw fieldType <name> "<rule>" <priority> [**-comment** "<string>"]
- save ns config

Example

The following example modifies the field type to add a comment.

```
set appfw fieldType SSN "^[1-9][0-9]{2,2}-[0-9}{2,2}-[0-9]{4,4}$" 1 -comment "US Social Security Number"
save ns config
```

To remove a field type by using the NetScaler command line

At the command prompt, type the following commands:

- rm appfw fieldType <name> "<url>"
- save ns config

Parameters for configuring a field type

name (Field Name)

The name of the field type. If you are adding a new field type, this can be a string of from one to 31 letters, numbers, and the underscore and hyphen symbols.

rule (Regular Expression)

The PCRE-format regular expression that defines the character format and length allowed in this field type.

url (URL)

A literal string or PCRE-format regular expression that describes the URL or URLs that host the web form(s) where the form field(s) are located.

priority (Priority)

A positive integer that designates the order in which this field type is checked for a match. A field type with a lower priority (such as 1) is checked before a field type with a higher priority (such as 2)

comment (Comments)

A comment. Optional.

To configure a field type by using the configuration utility

1. In the navigation pane, click **Application Firewall**.
2. In the details pane, under **Settings**, click **Manage Field Types**.
3. In the **Manage Field Types** dialog box, do one of the following:
   - To add a new field type to the list, click **Add**.
   - To change an existing field type, select the field type, and then click **Open**.

   The **Create Field Type** dialog box or the **Configure Field Type** dialog box appears.

   Note: If you select an existing field type designation and then click **Add**, the **Create Field Type** dialog box displays the information for that field type. You can modify that information to create your new field type.
4. In the dialog box, fill out the elements. They are:
   - Name
   - Regular Expression
   - Priority
   - Comment
5. Click **Create** or **OK**.
6. To remove a field type from the Field Types list, select the field type listing you want to remove, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding, modifying, and removing field types, click **Close**.

Examples

Following are some regular expressions for field types that you might find useful:

`^[1-9][0-9]{2,2}-[0-9}{2,2}-[0-9]{4,4}$` U.S. Social Security numbers

`^[A-C][0-9]{7,7}$` California driver's license numbers.

`^+[0-9]{1,3} [0-9() -]{1,40}$` International phone numbers with country codes.

`^[0-9]{5,5}-[0-9]{4,4}$` U.S. ZIP code numbers.

`^[0-9A-Za-z][0-9A-Za-z.+_-]{0,25}@([0-9A-Za-z][0-9A-Za-z_-]*[.]){1,4}[A-Za-z]{2`
Email addresses.

Logs, Statistics, and Reports

The information maintained in the logs and statistics, and displayed in the reports, provides important guidance for configuring and maintaining the application firewall.

The Application Firewall Logs

The logs provide information about the requests and responses that the application firewall has observed while protecting your web sites and applications. Most important, it logs each connection that matches a signature or a security check. You can observe the logs to determine which connections are matching a signature or security check. You can then use this information, along with your own knowledge about your protected web sites and applications, to determine whether the connections that each signature or check is matching are valid (*false positives*). If they are, you can either remove the signature or check from your configuration, or take appropriate measures to mitigate the false positives before you enable blocking for that signature or security check.

Each log contains the following fields:

- **Timestamp.** The date and time when the connection occurred.
- **Severity.** The severity level of the log.
- **Module.** The NetScaler module that generated the log entry.
- **Event Type.** The type of event, such as signature violation or security check violation.
- **Event ID.** The ID assigned to the event.
- **Client IP.** The IP address of the user whose connection was logged.
- **Transaction ID**. The ID assigned to the transaction that caused the log.
- **Session ID.** The ID assigned to the user session that caused the log.
- **Message.** The log message. Contains information identifying the signature or security check that triggered the log entry.

You can search on any of these fields, or any combination of information from different fields, to select logs to display, limited only by the capabilities of the tools you use to view the logs. You can observe the signatures by using the application firewall wizard to access the NetScaler syslog viewer, or manually by logging onto the NetScaler appliance or VPX virtual appliance.

- **Viewing by using the syslog viewer.** You invoke the syslog viewer from one of two locations: the **Select Signature Actions** page or the **Select Advanced Actions** page in the application firewall wizard. To invoke the syslog viewer for a signature, in the **Select Signature Actions** pane click the **logs** link to the right of that signature. To invoke the syslog viewer for a security check, in the **Select Advanced Actions** page,

security checks list, select that security check, and then beneath the list click the **Logs** button. Either procedure causes the configuration utility to download the current `ns.log` file and then display the entries that are relevant to that signature or security check.

The syslog viewer contains the following elements:

- *Module list box*. The NetScaler module whose logs you want to view. Always set to APPFW for application firewall logs.
- *Event Type list box*. The type of event. For signatures, this is always APPFW_SIGNATURE_MATCH. For security checks, this is the specific security check that you selected.
- *Severity*. Lets you specify only logs of a specific severity level. Leave blank to see all logs.
- *Find Now button*. Search the `nslog.file`, using the current criteria, and display the logs that match.
- *Clear button*. Resets your settings to the defaults.
- *Logs display window*. Displays the logs that meet the current criteria. Log information is displayed in several columns that correspond to the log information fields listed above. You can sort the display by clicking a column heading.
- *Log directory*. The directory where the logs are stored. If you have archived logs stored in a different directory and want to view those, you can click **Browse** and browse to that directory to display those logs in the Log files list.
- *Log files list*. A list of the log files in the Log directory. To download and uncompress an archived log file, select the file, and then click **Download**. To refresh the display, click **Refresh**.
- *Search in list box*. Searches in a particular section of logs when selecting logs to display in the Logs display window. To search something other than the log message, select a different choice.
- *Search string*. Search for the specified string or regular expression to choose the logs to display in the Logs display window. This field is filled out by the application firewall wizard for you with the appropriate value to display the logs relevant to the signature or security check that you selected. You can modify the string to choose logs based on different criteria.
- *Case Sensitive check box*. Select if the Search string is case sensitive.
- *Regular Expression check box*. Select if the Search string is a regular expression.
- *Clear button*. Resets the syslog viewer to its default settings.
- *Go button*. Uses the new search criteria to search the `ns.log` file and displays the results in the Logs display window.

For more information about the application firewall wizard, see The Application Firewall Wizard.

-

Viewing from the command line. Log onto the application firewall appliance, and then type the following command at the NetScaler command prompt:

```
shell
```

After the Unix shell is displayed, type the following command to navigate to the directory where the logs are stored:

```
cd /var/log
```

You can use the vi editor, or any Unix tool of your choice that you have installed on the application firewall appliance, to view the logs and filter the logs for specific entries.

The Application Firewall Statistics

When you enable the statistics action for application firewall signatures or security checks, the application firewall maintains information about connections that match that signature or security check. You can view the accumulated statistics information on the **Monitoring** tab of the main logon page of your application firewall appliance by selecting one of the following choices in the Select Group list box:

- **Application Firewall.** A summary of all statistics information gathered by your application firewall appliance for all profiles.
- **Application FIrewall (per profile).** The same information, but displayed per-profile rather than summarized.

You can use this information to monitor how your application firewall is operating and determine whether there is any abnormal activity or abnormal amounts of hits on a signature or security check. If you see such a pattern of abnormal activity, you can check the logs for that signature or security check, to diagnose the issue, and then take corrective action.

The PCI DSS Report

The Payment Card Industry (PCI) Data Security Standard (DSS), version 1.2, consists of twelve security criteria that most credit card companies require businesses who accept online payments via credit and debit cards to meet. These criteria are designed to prevent identity theft, hacking, and other types of fraud. If an internet service provider or online merchant does not meet the PCI DSS criteria, that ISP or merchant risks losing authorization to accept credit card payments through its web site.

ISPs and online merchants prove that they are in compliance with PCI DSS by having an audit conducted by a PCI DSS Qualified Security Assessor (QSA) Company. The PCI DSS report is designed to assist them both before and during the audit. Before the audit, it shows which application firewall settings are relevant to PCI DSS, how they should be configured, and (most important) whether your current application firewall configuration meets the standard. During the audit, the report can be used to demonstrate compliance with relevant PCI DSS criteria.

The PCI DSS report consists of a list of those criteria that are relevant to your application firewall configuration. Under each criterion, it lists your current configuration options,

indicates whether your current configuration complies with the PCI DSS criterion, and explains how to configure the application firewall so that your protected web site(s) will be in compliance with that criterion.

The PCI DSS report is located under **System** > **Reports**. To generate the report as an Adobe PDF file, click **Generate PCI DSS Report**. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

Note: To view this and other reports, you must have the Adobe Reader program installed on your computer.

The PCI DSS report consists of the following sections:

Description. A description of the PCI DSS Compliance Summary report.

Firewall License and Feature Status. Tells you whether the application firewall is licensed and enabled on your NetScaler appliance.

Executive Summary. A table that lists the PCI DSS criteria and tells you which of those criteria are relevant to the application firewall.

Detailed PCI DSS Criteria Information. For each PCI DSS criterion that is relevant to your application firewall configuration, the PCI DSS report provides a section that contains information about whether your configuration is currently in compliance and, if it is not, how to bring it into compliance.

Configuration. Data for individual profiles, which you access either by clicking **Application Firewall Configuration** at the top of the report, or directly from the **Reports** pane. The Application Firewall Configuration report is the same as the PCI DSS report, with the PCI DSS-specific summary omitted, and is described below.

The Application Firewall Configuration Report

The Application Firewall Configuration report is located under **System** > **Reports**. To display it, click **Generate Application Firewall Configuration Report**. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

The Application FIrewall Configuration report starts with a Summary page, which consists of the following sections:

- **Application Firewall Policies.** A table that lists your current application firewall policies, showing the policy name, the content of the policy, the action (or profile) it is associated with, and global binding information.

- **Application Firewall Profiles.** A table that lists your current application firewall profiles and indicates which policy each profile is associated with. If a profile is not associated with a policy, the table displays INACTIVE in that location.

To download all report pages for all policies, at the top of the Profiles Summary page click **Download All Profiles**. You display the report page for each individual profile by selecting that profile in the table at the bottom of the screen. The Profile page for an individual profile shows whether each check action is enabled or disabled for each check, and the other configuration settings for the check.

To download a PDF file containing the PCI DSS report page for the current profile, click **Download Current Profile** at the top of the page. To return to the **Profiles Summary** page, click **Application Firewall Profiles**. To go back to the main page, click **Home**. You can refresh the PCI DSS report at any time by clicking **Refresh** in the upper right corner of the browser. You should refresh the report if you make changes to your configuration.

Cache Redirection

In a typical deployment, different clients ask web servers for the same content repeatedly. To relieve the origin web server of processing each request, a NetScaler® appliance with cache redirection enabled can serve this content from a cache server instead of from the origin server.

The NetScaler analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers.

Cache redirection is a policy-based feature. By default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. For testing or maintenance, you might want to skip policy evaluation and direct all requests to the cache or to the origin server.

You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

A NetScaler configured for cache redirection can be deployed at the edge of a network, in front of the origin server, or anywhere along the network backbone. In an edge deployment, commonly used by Internet Service Providers (ISPs), cable companies, content delivery distribution networks, and enterprise networks, the NetScaler resides directly in front of the clients. In a server-side deployment, the NetScaler is closer to the origin servers.

Cache redirection is used most commonly with the HTTP service type, but it also supports the secure HTTPS protocol.

Cache Redirection Policies

A cache redirection virtual server applies cache redirection policies to each incoming request. By default, if a request matches one of the configured policies, it is considered non-cacheable, and the NetScaler appliance sends it to the origin server. Other requests are sent to a cache server. This behavior can be reversed, so that requests that match configured cache redirection policies are sent to cache servers.

The NetScaler provides a set of policies for cache redirection. If these built-in policies are not adequate for your deployment, you can configure user-defined cache redirection policies.

Note: Once you have determined which built-in cache redirection policies to use, or have created user-defined policies, proceed with configuring cache redirection. To use this feature, you must configure at least one cache redirection virtual server, and, for normal operation, you must bind at least one cache redirection policy to that virtual server.

Cache Redirection Policies

A cache redirection virtual server applies cache redirection policies to each incoming request. By default, if a request matches one of the configured policies, it is considered non-cacheable, and the NetScaler appliance sends it to the origin server. Other requests are sent to a cache server. This behavior can be reversed, so that requests that match configured cache redirection policies are sent to cache servers.

The NetScaler provides a set of policies for cache redirection. If these built-in policies are not adequate for your deployment, you can configure user-defined cache redirection policies.

Note: Once you have determined which built-in cache redirection policies to use, or have created user-defined policies, proceed with configuring cache redirection. To use this feature, you must configure at least one cache redirection virtual server, and, for normal operation, you must bind at least one cache redirection policy to that virtual server.

Built-in Cache Redirection Policies

The NetScaler appliance provides built-in cache redirection policies that handle typical cache requests. These policies are based on HTTP methods, the URL or URL tokens of the incoming request, the HTTP version, or the HTTP headers in the request and their values.

Built-in cache redirection policies can be directly bound to a virtual server and do not need further configuration.

Cache redirection policies use the simpler of two NetScaler expressions languages, called *classic expressions*. For a complete description of classic expressions and how to configure them, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

The NetScaler provides the following built-in cache redirection policies

built in Policy Name	Description
bypass-non-get	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, of if the HTTP request contains a pragma header.
bypass-dynamic-url	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: • cgi • asp • exe • cfm • ex • shtml • htx Also bypass the cache if the URL starts with any of the following: • /cgi-bin/ • /bin/ • /exec/

bypass-urltokens	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie	Bypass the cache for any URL that has a cookie header and an extension other than .gif or .jpg.

Displaying the Built-in Cache Redirection Policies

You can display the available cache redirection polices by using the NetScaler command line or the configuration utility.

To display the built-in cache redirection policies by using the NetScaler command line

At the NetScaler command prompt, type:

show cr policy [<policyName>]

Example

```
> show cr policy
1)      Cache-By-Pass RULE: NS_NON_GET          Policy:bypass-non-get
2)      Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE || NS_CACHECONTROL_NOCACHE || NS_HEADER_
3)      Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE || NS_EXT_CFM || NS_EXT_EX || N
4)      Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-urltokens
5)      Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF && NS_EXT_NOT_JPEG)     Policy:byp
 Done
>
```

To display the built-in cache redirection policies by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Policies**. The configured cache redirection policies appear in the details pane.
2. Select one of the configured policies to view details.

Configuring a Cache Redirection Policy

A cache redirection policy includes one or more expressions (also called *rules*). Each expression represents a condition that is evaluated when the client request is compared to the policy.

You do not explicitly configure actions for cache redirection policies. By default, the NetScaler considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection uses the *classic policy* format, which is described in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673. Each policy has a name and includes an expression or a set of expressions that are combined by using logical operators.

To add a cache redirection policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a cache redirection policy and verify the configuration:

- add cr policy <policyName> **-rule** <expression>
- show cr policy [<policyName>]

Examples

Policy with a simple expression:

```
> add cr policy Policy-CRD-1 -rule "REQ.HTTP.URL != /*.jpeg"
 Done
> show cr policy Policy-CRD-1
        Cache-By-Pass RULE: REQ.HTTP.URL != '/*.jpeg'   Policy:Policy-CRD-1
 Done
>
```

Policy with a compound expression:

```
> add cr policy Policy-CRD-2 -rule "REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == /*.cgi || REQ.HTTP.U
```

```
 Done
> show cr policy Policy-CRD-2
        Cache-By-Pass RULE: REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == '/*.cgi' || REQ.HTTP.URL != '/
 Done
>
```

Policy that evaluates a header:

```
> add cr policy Policy-CRD-3 -rule "REQ.HTTP.HEADER If-Modified-Since EXISTS"
 Done
> show cr policy Policy-CRD-3
        Cache-By-Pass RULE: REQ.HTTP.HEADER If-Modified-Since EXISTS    Policy:Policy-CRD-3
 Done
>
```

To modify or remove a cache redirection policy by using the NetScaler command line

- To modify a cache redirection policy, use the set cr policy command, which is just like using the add cr policy command, except that you enter the name of an existing policy.
- To remove a policy, use the rm cr policy command, which accepts only the <name> argument. You can only remove a cache redirection policy that is not bound to a cache redirection virtual server. If the policy is bound to a virtual server, you must first unbind the policy, and then remove it from the NetScaler.

For details on unbinding a cache redirection policy, see Unbinding a Policy from a Cache Redirection Virtual Server.

Parameters for creating a cache redirection policy

policyName

Name of the cache redirection policy. This is a mandatory parameter, and the value cannot be changed after the policy is created.

rule

An expression that the NetScaler evaluates to identify non-cacheable requests. Can consist of multiple expressions joined by AND and OR operators.

To configure a cache redirection policy with a simple expression by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create Cache Redirection Policy** dialog box, in the **Name*** text box, type the name of the policy, and then in the **Expression** area, click **Add**.
4. To configure a simple expression, enter the expression. Following is an example of an expression that checks for a .jpeg extension in a URL:
   - **Expression Type**-General
   - **Flow Type** -REQ
   - **Protocol** -HTTP
   - **Qualifier** -URL
   - **Operator** - !=
   - **Value*** - /*.jpeg

   The simple expression in the following example checks for an If-Modified-Since header in a request:
   - **Expression Type** -General
   - **Flow Type** -REQ
   - **Protocol** -HTTP
   - **Qualifier** -HEADER
   - **Operator** -EXISTS
   - **Header Name** -If-Modified-Since
5. When you are finished entering the expression, click **OK** or **Create**, and then click **Close**.

To configure a cache redirection policy with a compound expression by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Name** text box, enter a name for the policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of content this policy was created to detect.
4. Choose the type of compound expression that you want to create. Your choices are:
   - **Match Any Expression**. The policy matches the traffic if one or more individual expressions match the traffic.
   - **Match All Expressions**. The policy matches the traffic only if every individual expression matches the traffic.
   - **Tabular Expressions**. Switches the Expressions list to a tabular format with three columns. In the rightmost column, you place one of the following operators:
     - The AND [&&] operator, to require that, to match the policy, a request match both the current expression and the following expression.
     - The OR [||] operator, to require that, to match the policy, a request match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.

     You can also group expressions in nested subgroups by selecting an existing expression and clicking one of the following operators:
     - The BEGIN SUBGROUP [+ ( ] operator, which tells the NetScaler appliance to begin a nested subgroup with the selected expression. (To remove this operator from the expression, click -(.)
     - The END SUBGROUP [+)] operator, which tells the NetScaler appliance to end the current nested subgroup with the selected expression. (To remove this operator from the expression, click -) .)
   - **Advanced Free-Form**. Switches off the Expressions Editor entirely and turns the Expressions list into a text area in which you type your compound expression. This is both the most powerful and the most difficult method of creating a policy expression, and is recommended only for those thoroughly familiar with the NetScaler classic expressions language.

     For more information about creating classic expressions in the Advanced Free-Form text area, see "Configuring Classic Policies and Expressions" in the *Citrix NetScaler*

Policy Configuration and Reference Guide at http://support.citrix.com/article/CTX128673.

Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that is what you want to do.

5. If you chose **Match Any Expression**, **Match All Expressions**, or **Tabular Expressions**, click Add to display the **Add Expression** dialog box.

   You should leave the expression type set to General for cache redirection policies.

6. In the **Flow Type** drop-down list, choose a flow type for your expression.

   The flow type determines whether the policy examines incoming or outgoing connections. You have two choices:

   - **REQ**. Configures the NetScaler appliance to examine incoming connections, or requests.
   - **RES**. Configures the appliance to examine outgoing connections, or responses.

7. In the **Protocol** drop-down list, choose a protocol for your expression.

   The protocol determines the type of information that the policy examines in the request or response. Depending upon whether you chose REQ or RES in the previous drop-down list, either all four of the following choices, or three of them, are available:

   - **HTTP**. Configures the appliance to examine the HTTP header.
   - **SSL**. Configures the appliance to examine the SSL client certificate. Available only if you chose REQ (requests) in the previous drop-down list.
   - **TCP**. Configures the appliance to examine the TCP header.
   - **IP**. Configures the appliance to examine the source or destination IP.

8. Choose a qualifier for your expression from the **Qualifier** drop-down list.

   The contents of the **Qualifier** drop-down list depend on which protocol you chose. The following table describes the choices available for each protocol.

   Table 1. Cache Redirection Policy Qualifiers Available for Each Protocol

Protocol	Qualifier	Definition
HTTP	METHOD	The HTTP method used in the request.

	URL	The contents of the URL header.
	URLTOKENS	The URL tokens in the HTTP header.
	VERSION	The HTTP version of the connection.
	HEADER	The header portion of the HTTP request.
	URLLEN	The length of the contents of the URL header.
	URLQUERY	The query portion of the contents of the URL header.
	URLQUERYLEN	The length of the query portion of the URL header.
SSL	CLIENT.CERT	The SSL client certificate as a whole.
	CLIENT.CERT.SUBJECT	The contents of the client certificate subject field.
	CLIENT.CERT.ISSUER	The client certificate issuer.
	CLIENT.CERT.SIGALGO	The signature algorithm used in the client certificate.
	CLIENT.CERT.VERSION	The client certificate version.
	CLIENT.CERT.VALIDFROM	The date from which the client certificate is valid. (The start date.)
	CLIENT.CERT.VALIDTO	The date after which the client certificate is no longer valid. (The end date.)
	CLIENT.CERT.SERIALNUMBER	The client certificate serial number.
	CLIENT.CIPHER.TYPE	The encryption method used in the client certificate.
	CLIENT.CIPHER.BITS	The number of significant bits in the encryption key.
	CLIENT.SSL.VERSION	The SSL version of the client certificate.

TCP	SOURCEPORT	The source port of the TCP connection.
	DESTPORT	The destination port of the TCP connection.
	MSS	The maximum segment size (MSS) of the TCP connection.
IP	SOURCEIP	The source IP of the connection.
	DESTIP	The destination IP of the connection.

9. Choose the operator for your expression from the **Operator** drop-down list.

   Your choices depend on the qualifier you chose in the previous step. The complete list of operators that can appear in this drop-down list is:

   - == . Matches the following text string exactly.
   - != . Does not match the following text string.
   - > . Is greater than the following integer.
   - CONTAINS . Contains the following text string.
   - CONTENTS . The contents of the designated header, URL, or URL query.
   - EXISTS . The specified header or query exists.
   - NOTCONTAINS . Does not contain the following text string.
   - NOTEXISTS . The specified header or query does not exist.

   If you want this policy to operate on requests sent to a specific Host, you can leave the default, the equals (==) sign.

10. If the **Value** text box is visible, type the appropriate string or number into the text box.

    For example, if you want this policy to select requests sent to the host `shopping.example.com`, you would type that string in the **Value** text box.

11. If you chose HEADER as the qualifier, type the header you want in the **Header Name** text box.

12. Click **OK** to add your expression to the **Expression** list.

13. Repeat steps 4 through 11 to create additional expressions.

14. Click **Close** to close the **Add Expression** dialog box and return to the **Create Cache Redirection Policy** dialog box.

Cache Redirection Configurations

Depending on your deployment and network topology, you can configure one of the following types of cache redirection:

- **Transparent**. A transparent cache can reside on a variety of points along a network backbone to alleviate traffic along the delivery route. In transparent mode, the cache redirection virtual server intercepts all traffic flowing to the NetScaler appliance and applies cache redirection policies to determine whether content should be served from the cache or from the origin server.

- **Forward proxy**. A forward proxy cache server resides on the edge of an enterprise LAN and faces the WAN. In the forward proxy mode, the cache redirection virtual server resolves the hostname of the incoming request by using a DNS server and forwards requests for non-cacheable content to the resolved origin servers. Cacheable requests are sent to the configured cache servers.

- **Reverse proxy**. Reverse proxy caches are configured for specific origin servers. Incoming traffic directed to the reverse proxy, can either be served from a cache server or be sent to the origin server with or without modification to the URL.

Configuring Transparent Redirection

When you configure transparent cache redirection, the NetScaler appliance evaluates all traffic it receives, to determine whether it is cacheable. This mode alleviates traffic along the delivery route and is often used when the cache server resides on the backbone of an ISP or carrier.

By default, cacheable requests are sent to a cache server, and non-cacheable requests to the origin server. For example, when the NetScaler appliance receives a request that is directed to a web server, it compares the HTTP headers in the request with a set of policy expressions. If the request does not match the policy, the appliance forwards the request to a cache server. If the response does match a policy, the appliance forwards the request, unchanged, to the web server.

For details on how to modify this default behavior, see Directing Policy Hits to the Cache instead of the Origin.

To configure transparent redirection, first enable cache redirection and load balancing, and configure edge mode. Then, create a cache redirection virtual server with a wildcard IP address (*), so that this virtual server can receive traffic coming to the NetScaler on any IP address the appliance owns. To this virtual server, bind cache redirection policies that describe the types of requests that should not be cached. Then, create a load balancing virtual server that will receive traffic from the cache redirection virtual server for cacheable requests. Finally, create a service that represents a physical cache server and bind it to the load balancing virtual server.

Enabling Cache Redirection and Load Balancing

The NetScaler cache redirection and load balancing features are not enabled by default. They must be enabled before any cache redirection configuration can take effect.

To enable cache redirection and load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following command to enable cache redirection and load balancing and verify the settings:

- enable ns feature cr lb
- show ns feature

Example

```
> enable ns feature cr lb
 Done
> show ns feature

      Feature                 Acronym          Status
      -------                 -------          ------
 1)   Web Logging               WL               ON
 2)   Surge Protection          SP               ON
 3)   Load Balancing            LB               ON
 4)   Content Switching          CS               ON
 5)   Cache Redirection          CR               ON
 6)   Sure Connect

         ...
         ...
         ...

 23)   HTML Injection            HTMLInjection       ON
 24)   NetScaler Push            push                OF
 Done
>
```

To enable cache redirection and load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. To enable cache redirection, in the details pane, under **Modes and Features**, click **Configure advanced features**.
    a. In **Configure Advanced Features** dialog box, select the check box next to the **Cache Redirection**, and then click **OK**.
    b. In **Enable/Disable Feature(s)?** dialog box, click **Yes**.
3. To enable load balancing, in the details pane, under **Modes and Features**, click **Configure basic features**.
    a. In **Configure Basic Features** dialog box, select the check box next to the **Load Balancing**, and then click **OK**.
    b. In **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Configuring Edge Mode

When deployed at the edge of a network, the NetScaler appliance dynamically learns about the servers on that network. Edge mode enables the appliance to dynamically learn about up to 40,000 HTTP servers and proxy TCP connections for these servers.

This mode turns off collection of statistics for the dynamically learned services and is typically used in transparent deployments for cache redirection.

To enable edge mode by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable edge mode and verify the setting:

- enable ns mode edge
- show ns mode

Example

```
> enable ns mode edge
 Done

> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
        ...
        ...
        ...
 6)   MAC-based forwarding       MBF             ON
 7)   Edge configuration         Edge            ON
 8)   Use Subnet IP              USNIP           OFF
        ...
        ...
        ...
 16)  Bridge BPDUs               BridgeBPDUs     OFF
 Done
>
```

Parameters for enabling edge mode

Mode

The name of the mode to be enabled. This is a mandatory argument.

To enable edge mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Configure modes**.
3. In **Configure Modes** dialog box, select the check box next to the **Edge Configuratio**, and then click **OK**.
4. In **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Configuring a Cache Redirection Virtual Server

By default, a cache redirection virtual server forwards cacheable requests to the load balancing virtual server for the cache, and forwards non-cacheable requests to the origin server (except in a reverse proxy configuration, in which non-cacheable requests are sent to a load balancing virtual server). There are three types of cache redirection virtual servers: transparent, forward proxy, and reverse proxy.

A transparent cache redirection virtual server uses an IP address of * and a port number, usually 80, that can accept HTTP traffic sent to any IP address that the NetScaler represents. As a result, you can configure only one transparent cache redirection virtual server. Any additional cache redirection virtual servers that you configure must be forward proxy or reverse proxy redirection servers.

To add a cache redirection virtual server in transparent mode by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a cache redirection virtual server and verify the configuration:

- add cr vserver <name> <serviceType> [<IPAddress> <port>] [**-cacheType** <cacheType>] [**-redirect** <redirect>] [**-cacheVserver** <string>]
- show cr vserver [<name>]

Example

```
add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect POLICY -cacheVserver Vserver-L
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: RULE       Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
 Done
>
```

To modify or remove a cache redirection virtual server by using the NetScaler command line

- To modify a virtual server, use the set cr vserver command, which is just like using the add cr vserver command, except that you enter the name of an existing virtual server.
- To remove a virtual server, use the rm cr vserver command, which accepts only the <name> argument.

Parameters for adding a cache redirection virtual server

name

The name of the virtual server you are configuring. The name can begin with a letter, a number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the type of service being added. (Cannot be changed after the service has been created.)

serviceType

The type of data that the virtual server will handle. For typical web traffic, the protocol is HTTP. Possible Values: HTTP, SSL.

IPAddress

The IP address of the virtual server that you are adding. For transparent cache redirection, assign an asterisk (*) as the IP address, so that the virtual server listens on all IP addresses configured on the NetScaler.

port

The port number on which the virtual server receives traffic.

cacheType

The type of cache redirection you are configuring. Possible values: TRANSPARENT, REVERSE, FORWARD.

redirect

The redirect method to be used. Possible values: CACHE, ORIGIN, POLICY.

For normal operation, specify POLICY based redirection and bind appropriate cache redirection policies to the virtual server.

cacheVserver

The name of a load balancing virtual server to which this redirection virtual server sends cache requests.

To add a cache redirection virtual server in transparent mode by using the configuration utility

1. In the navigation pane, click **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Cache Redirection)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for adding a cache redirection virtual server" as shown:
   - **Name***—name
   - **Port***—port

   * A required parameter
4. In the **Protocol** drop-down list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the standard port for the selected protocol, enter a new value in the **Port** field.
5. Click the **Advanced** tab.
6. Verify that **Cache Type** is set to TRANSPARENT and **Rediret** is set to POLICY.
7. If you have already configured load balancing virtual servers for cache redirction, select a virtual server from the **Cache Server** drop-down list.
8. Click **Create**, and then click **Close**. The **Cache Redirection Virtual Servers** pane displays the new virtual server.
9. Select the new cache redirection virtual server to display the details of its configuration.

Binding Policies to the Cache Redirection Virtual Server

Cache redirection policies are not automatically bound to the cache redirection virtual server. A policy based cache redirection virtual server cannot function unless you bind at least one policy to it.

To bind policies to a cache redirection virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

- bind cr vserver <name> **-policyName** <string>
- show cr vserver [<name>]

Example

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
 Done

> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: RULE       Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: bypass-dynamic-url
3)     Cache bypass  Policy: bypass-urltokens
4)     Cache bypass  Policy: bypass-cookie
 Done
```

>

Parameters for binding policies to a cache redirection virtual server

name

The name of the cache redirection virtual server you are binding the policy to.

policyName

The name of the policy being bound to the cache redirection virtual server. The policy must already be created on the NetScaler appliance before it is bound.

To bind a user-defined policy to a cache redirection virtual server by using the configuration utility

1. In the navigation pane, expand **Cache Redirection** and click **Virtual Servers**.
2. Click the virtual server that you want to configure, and click **Open**.
3. On the **Policies** tab, select type of the policy and then click **Insert Policy**.
4. Under **Policy Name** column, select the policy that you want to bind.
5. Click **OK**.

Unbinding a Policy from a Cache Redirection Virtual Server

When you unbind a policy from the cache redirection virtual server, the NetScaler appliance no longer applies the policy when evaluating client requests.

To unbind a policy from a cache redirection virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

- unbind cr vserver <name> **-policyName** <string>
- show cr vserver [<name>]

Example

```
unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: RULE       Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
 Done
>
```

To unbind a user-defined policy from a cache redirection virtual server by using the configuration utility

1. In the navigation pane, expand **Cache Redirection** and click **Virtual Servers**.
2. Click the virtual server that you want to configure, and then click **Open**.
3. On the **Policies** tab, under **Policy Name**, select the policy that you want to unbind.
4. Click **Unbind Policy**, and then click **OK**.

Creating a Load Balancing Virtual Server

The cache redirection virtual server on the NetScaler appliance can send requests to either a cache server farm, if the request is cacheable, or to the origin server farm if the request is not cacheable.

Each cache server is represented on the appliance by a service, which is bound to a load balancing virtual server that receives requests from the cache redirection virtual server and forwards those requests to the servers.

For details on configuring load balancing virtual servers and other configuration options, see Load Balancing.

To create a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a load balancing virtual server and verify the configuration:

- add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
- show lb vserver [<name>]

Example

```
> add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:00:08.470
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
```

```
      Push Label Rule: none
 Done
>
```

Parameters for creating a load balancing virtual server

name

The name of the virtual server being created.

serviceType

The type of traffic the virtual server will be receiving. For all web traffic, create virtual servers of type HTTP or, for secure Web traffic, HTTPS.

IPAddress

The IP address of the virtual server. This is the IP address that will receive incoming traffic.

port

The port number on which the incoming traffic will be received.

To create a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a load balancing virtual server" as shown:
   - **Name***-name
   - **IP Address***- IPAddress
   - **Port***-port

   * A required parameter
4. In the **Protocol*** drop down list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the well-known port for the selected protocol, enter a new value in the **Port** field.
5. Click **Create**, and then click **Close**. The **Load Balancing Virtual Servers** pane displays the new virtual server.

Configuring an HTTP Service

On the NetScaler appliance, a service represents a physical server on the network. In the transparent cache redirection configuration, the service represents the cache server. Cacheable requests are sent by the cache redirection virtual server to the load balancing virtual server, which in turn forwards each request to the correct service, which passes it on to the cache server.

To configure an HTTP service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an HTTP service and verify the configuration:

- add service <name> <IP> <serviceType> <port> **-cacheType** <cacheType>
- show service [<name>]

Example

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType TRANSPARENT
 Done
> show service Service-HTTP-1
      Service-HTTP-1 (10.102.29.40:80) - HTTP
      State: DOWN
      Last state change was at Fri Jul  2 09:14:17 2010
      Time since last state change: 0 days, 00:00:13.820
      Server Name: 10.102.29.40
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): YES
      Idle timeout: Client: 180 sec   Server: 360 sec
      Client IP: DISABLED
      Cache Type: TRANSPARENT Redirect Mode:
      Cacheable: NO
      SC: OFF
      SP: ON
      Down state flush: ENABLED

1)     Monitor Name: tcp-default
            State: DOWN     Weight: 1
```

```
            Probes: 3       Failed [Total: 3 Current: 3]
            Last response: Failure - Time out during TCP connection establishment stage
            Response Time: N/A
 Done
>
```

To modify or remove a service by using the NetScaler command line

- To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service.
- To remove a service, use the rm service command, which accepts only the <name> argument.

Parameters for adding an HTTP service

name

The name of the service you are configuring. The name can begin with a letter, a number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the type of service being added. (Cannot be changed after the service has been created.)

IP

The physical IP address of the server that the service you are configuring represents. Make sure that the server is reachable by the NetScaler.

port

The port number on which the service sends and receives data to and from the server.

serviceType

The type of data that will be transferred between the NetScaler and the server. Typically, for web server caches, the serviceType is HTTP.

cacheType

The type of cache redirection you are configuring.

To add an HTTP service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for adding an HTTP service" as shown:
   - **Service Name***—name
   - **Server***— IP
   - **Port***—port

   * A required parameter
4. In the **Protocol*** drop-down list, select a supported protocol (for example, **HTTP**).
5. Click **Create**, and then click **Close**.

Binding/Unbinding a Service to/from a Load Balancing Virtual Server

You must bind a service to the load balancing virtual server. This enables the load balancer to forward the request to the server that the service represents. If your configuration changes, you can unbind a service from the load balancing virtual server.

To bind a service to a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

- bind lb vserver <name> <serviceName>
- show lb vserver [<name>]

Example

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:42:25.610
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
>
```

To unbind a service from a load balancing virtual server by using the NetScaler command line

To unbind a service, use the unbind lb vserver command instead of bind lb vserver.

Parameters for binding/unbinding a service to/from a load balancing virtual server

name

The virtual server name from which the service will be bound/unbound. This is a mandatory argument. Maximum Length: 127

serviceName

The service name (created with the addService command) that will be unbound. Maximum Length: 127

To bind/unbind a service from a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server from which you want to bind/unbind the service, and then click **Open**.
3. On the **Services** tab, in the **Active** column, select/clear the check box next to the **Service Name**.
4. Click **OK**.

Assigning a Port Range to the NetScaler

Sharing of the client IP address may create a conflict that makes network devices, such as routers, cache servers, origin servers, and other NetScaler appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

A method to solve this problem is to assign a source port range to the NetScaler appliance. This allotment enables network devices to unambiguously identify the NetScaler appliance that sent the request.

To assign a source port range to a NetScaler appliance by using the NetScaler command line

At the NetScaler command prompt, type:

set ns config **-crPortRange** <startPortNumber-endPortNumber>

Parameters for assigning source port range

startPortNumber

Lower limit of the port range

endPortNumber

Upper limit of the port range

The range should be from 1024 through 65535.

To assign a source port range to a NetScaler appliance by using the NetScaler configuration utility

1. In the navigation pane, click **System**, and then click **Settings**.
2. In the **Settings** group, click the **Change global system settings** link.
3. In the **Cache Redirection Port Range** group, specify the port range for the NetScaler by typing a port number for **Start Port** and a port number for **End Port**.
4. Click **OK**.

Enabling Load Balancing Virtual Servers to Redirect Requests to Cache

If a load balancing virtual server is configured to listen on a particular IP address and port combination, it takes precedence over the cache redirection virtual server for any requests destined for that address-port combination. Therefore, the cache redirection virtual server does not process those requests.

If you want to override this functionality and let the cache redirection virtual server decide whether the request should be served from the cache or not, configure the particular load balancing virtual server to be cacheable.

Such a configuration is typically used when an ISP uses a NetScaler appliance at the edge of its network and all traffic flows through the appliance.

To enable load balancing virtual servers to redirect requests to the cache by using the NetScaler command line

At the NetScaler command prompt, type:

- set lb vserver <name> [**-cacheable** (YES | NO)]
- show lb vserver [<name>]

Example

```
set lb vserver Vserver-LB-CR –cacheable YES
> show lb vserver vserver-LB-CR
      Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
      State: DOWN
      Last state change was at Fri Jul  2 08:47:52 2010
      Time since last state change: 0 days, 01:05:51.510
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  1 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
```

```
    Persistence: NONE
    Cacheable: YES  PQ: OFF SC: OFF
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
```

For transparent cache redirection, the NetScaler intercepts all traffic and evaluates every request to determine whether it is cacheable. Non-cacheable requests are sent unchanged to the origin server.

When using transparent cache redirection, you may want to turn off cache redirection for load balancing virtual servers that always direct traffic to origin servers.

To turn off caching for a load balancing virtual server by using the NetScaler command line

To turn off caching for a load balancing virtual, use the unset lb vserver command instead of set lb vserver. Specify a value of NO value for the **-cacheable** parameter**.**

To enable or disable load balancing virtual servers to redirect requests to the cache by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server from which you want to enable/disable the caching, and then click **Open**.
3. On the **Advanced tab**, select/clear **Cache Redirection** check box.
4. Click **OK**.

Configuring Forward Proxy Redirection

A forward proxy is a single point of contact for a client or group of clients. In this configuration, the NetScaler appliance redirects non-cacheable requests to an origin server and redirects cacheable requests to either a forward proxy cache or a transparent cache.

When the NetScaler is configured as a forward proxy, users must modify their browsers so that the browser sends requests to the forward proxy instead of the destination servers.

A forward proxy cache redirection virtual server on the NetScaler compares the request with a policy for caching. If the request is not cacheable, the NetScaler queries a DNS load balancing virtual server for resolution of the destination, and then sends the request to the origin server. If the request is cacheable, the NetScaler forwards the request to a load balancing virtual server for the cache.

The NetScaler relies on a host domain name or IP address in the request's HOST header to determine the requested destination. If there is no HOST header in the request, the appliance inserts a HOST header based on the destination IP address in the request.

Typically, the NetScaler appliance acts as a forward proxy in an enterprise LAN. In such a configuration, the appliance resides at the edge of an enterprise LAN and intercepts client requests before they are fanned out to the WAN. Configuring the appliance in the forward proxy mode reduces traffic on the WAN.

To configure forward proxy cache redirection, first enable load balancing and cache redirection on the NetScaler. Then, configure a DNS load balancing virtual server and associated services. Also configure a load balancing virtual server and bind to it appropriate services for the cache. Configure a forward proxy cache redirection virtual server and bind the DNS and load balancing virtual servers to it. You must also configure caching policies and bind them to the cache redirection virtual server. To complete the setup, configure the client browsers to use the forward proxy.

For details on how to enable cache redirection and load balancing on the NetScaler, see Enabling Cache Redirection and Load Balancing.

For details on how to create a load balancing virtual server, see Creating a Load Balancing Virtual Server.

For details on how to configure services that represent the cache server, see Configuring an HTTP Service.

For details on how to bind the service to a virtual server, see Binding Services to the Virtual Server.

For details on how to create a forward proxy cache redirection server, see Configuring a Cache Redirection Virtual Server, and create a virtual server of type TRANSPARENT or FORWARD.

For details on binding cache redirection policies to the cache redirection virtual server, see Configuring a Cache Redirection Policy.

Creating a DNS Service

A DNS service is a representation, on the NetScaler appliance, of a physical DNS server in the network. A DNS load balancing virtual server sends DNS requests to the DNS server in the network through such a service.

To create a DNS service by using the NetScaler command line

At the NetScaler command line, type the following commands to create a DNS service and verify the configuration :

- add service <name> <IP> <serviceType> <port>
- show service [<name>]

Example

```
add service Service-DNS-1 10.102.29.41 DNS 53
show service Service-DNS-1
	Service-DNS-1 (10.102.29.41:53) - DNS
	State: DOWN
	Last state change was at Fri Jul  2 10:14:32 2010
	Time since last state change: 0 days, 00:00:13.550
	Server Name: 10.102.29.41
	Server ID : 0   Monitor Threshold : 0
	Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
	Use Source IP: NO
	Client Keepalive(CKA): NO
	Access Down Service: NO
	TCP Buffering(TCPB): NO
	HTTP Compression(CMP): NO
	Idle timeout: Client: 120 sec   Server: 120 sec
	Client IP: DISABLED
	Cacheable: NO
	SC: OFF
	SP: OFF
	Down state flush: ENABLED

1)	Monitor Name: ping-default
		State: DOWN     Weight: 1
		Probes: 3       Failed [Total: 3 Current: 3]
		Last response: Failure - Probe timed out.
		Response Time: 2000.0 millisec
 Done
```

Parameters for creating a DNS service

name

The name of the service you are configuring. The name can begin with a letter, a number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the type of service being added. (Cannot be changed after the service has been created.)

IP

The physical IP address of the server that the service you are configuring represents. Make sure that the server is reachable by the NetScaler.

port

The port number on which the service sends and receives data to and from the server.

serviceType

The type of data that will be transferred between the NetScaler and the server. Typically, for DNS service, the serviceType is DNS.

To add an DNS service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a DNS service" as shown:
   - **Service Name***—name
   - **Server***—IP
   - **Port***—port

   * A required parameter
4. In the **Protocol*** drop down list, select a supported protocol (for example, **DNS**).
5. Click **Create**, and then click **Close**.

Creating a DNS Load Balancing Virtual Server

The DNS virtual server enables the forward proxy to perform DNS resolution before forwarding a client request to an origin server. The DNS load balancing virtual server is associated with the DNS service that represents the physical DNS server on the network.

To create a DNS load balancing virtual server by using the NetScaler command line

At the NetScaler command line, type the following commands to create a DNS load balancing virtual server and verify the configuration:

- add lb vserver <name> <serviceType>
- show lb vserver [<name>]

Example

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:00:08.10
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
 Done
>
```

Parameters for creating a DNS load balancing virtual server

name

The name of the virtual server being created.

serviceType

The type of traffic the virtual server will be receiving.

To create a DNS load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, in the **Name** box, type a name for the virtual server.
4. In the **Protocol*** drop down list, select a supported protocol (for example, **DNS**).
5. Click **Create**, and then click **Close**. The **DNS Virtual Servers** pane displays the new virtual server.

Binding the DNS Service to the Virtual Server

For the DNS server to respond to DNS requests, the service representing the DNS server must be bound to the DNS virtual server.

To bind the DNS service to the load balancing virtual server:

At the NetScaler command prompt, type the following commands to bind the DNS service to the load balancing virtual server and verify the configuration:

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example

```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:12:16.80
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
```

To unbind a DNS service from the load balancing virtual server:

Use the unbind lb vserver command instead of bind lb vserver.

Parameters for Binding/Unbinding a DNS service to/from a load balancing virtual server

name

The name of the virtual server to/from which the service will be bound/unbound. This is a mandatory argument. Maximum Length: 127

serviceName

The name of the service (created with the addService command) that will be bound or unbound. Maximum Length: 127

To Bind/Unbind a DNS service to/from a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to/from which you want to bind/unbind the DNS service, and then click **Open**.
3. On the **Services** tab, in the **Active** column, select/clear the check box next to the **Service Name**.
4. Click **OK**.

Configuring a Client Web Browser to Use a Forward Proxy

When you configure the NetScaler appliance as forward proxy cache redirection virtual server in the network, you must configure the client Web browser to send requests to the forward proxy. Typically, when you use a forward proxy, the only route to the servers in the network is through the forward proxy.

Refer the documentation for your browser to configure the browser to use a forward proxy. Specify the IP address and port number of the forward proxy cache redirection virtual server for this configuration.

Configuring Reverse Proxy Redirection

A reverse proxy resides in front of one or more Web servers and shields the origin server from client requests. Often, a reverse proxy cache is a front-end for all client requests to a server. An administrator assigns a reverse proxy cache to a specific origin server. This is unlike transparent and forward proxy caches, which cache frequently requested content for all requests to any origin server, and the choice of a server is based on the request.

Unlike a transparent proxy cache, the reverse proxy cache has its own IP address and can replace destination domains and URLs in a non-cacheable request with new destination domains and URLs.

You can deploy reverse proxy cache redirection at the origin-server side or at the edge of a network. When deployed at the origin server, the reverse proxy cache redirection virtual server is a front-end for all requests to the origin server.

In the reverse proxy mode, when the NetScaler receives a request, a cache redirection virtual server evaluates the request and forwards it to either a load balancing virtual server for the cache or a load balancing virtual server for the origin. The incoming request can be transformed by changing the host header or the host URL before they it is sent to the backend server.

To configure reverse proxy cache redirection, first enable cache redirection and load balancing. Then, configure a load balancing virtual server and services to send cacheable requests to the cache servers. Also configure a load balancing virtual server and associated services for the origin servers. Then, configure a reverse proxy cache redirection virtual server and bind relevant cache redirection policies to it. Finally, configure mapping policies and bind them to the reverse proxy cache redirection virtual server.

The mapping policies have an associated action that enables the cache redirection virtual server to forward any non-cacheable request to the load balancing virtual server for the origin.

Be sure to create the default cache server destination.

For details on how to enable cache redirection and load balancing on the NetScaler, see Enabling Cache Redirection and Load Balancing.

For details on how to create a load balancing virtual server, see Creating a Load Balancing Virtual Server.

For details on how to configure services that represent the cache server, see Configuring an HTTP Service.

For details on how to bind the service to a virtual server, see Binding Services to the Virtual Server.

For details on how to create a reverse proxy cache redirection server, see Configuring a Cache Redirection Virtual Server, and create a virtual server of type REVERSE.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Binding Policies to the Cache Redirection Virtual Server.

Configuring Mapping Policies

If an incoming request is non-cacheable, the reverse-proxy cache redirection virtual server replaces the domain and URL in the request with the domain and URL of a target origin server and forwards the request to the load balancing virtual server for the origin.

A mapping policy enables the reverse proxy cache redirection virtual server to replace the destination domain and URL and forward the request to the load balancing virtual server for the origin.

A mapping policy must first translate the domain and the URL, and then pass the request on to the origin load balancing virtual server.

A mapping policy can map a domain, a URL prefix, and a URL suffix, as follows:

- Domain mapping: You can map a domain without a prefix or suffix. The domain mapping is the default mapping for the virtual server (for example, mapping www.mycompany.com to www.myrealcompany.com).
- Prefix mapping: You can replace a specified pattern prefixed as part of the URL (for example, mapping www.mycompany.com/sports/index.html to www.mycompany.com/news/index.html).
- Suffix mapping: You can replace the file suffix in the URL (for example, mapping www.mycompany.com/sports/index.html to www.mycompany.com/sports/index.asp).

The source and the destination strings being mapped must be similar. If you specify a source domain, you must specify a destination domain, and if you specify a source suffix, you must specify a destination suffix. Similarly, if you specify an exact URL from the source, the target URL must also be an exact URL.

Once you configure mapping policies for the reverse proxy mode, you must bind them to the cache redirection virtual server.

You can use combinations of the source URL, target URL, and source and target domains to configure all three types of domain mapping.

To configure a mapping policy for reverse proxy mode by using the NetScaler command line

At the NetScaler command prompt, type the following command to add a policy map and verify the configuration:

- add policy map <mapPolicyName> **-sd** <string> [**-su** <string>] [**-td** <string>] [**-tu** <string>]
- show policy map [<mapPolicyName>]

Example

The following command maps a domain in a client request to a target domain:

```
> add policy map myMappingPolicy -sd www.mycompany.com -td www.myrealcompany.com
 Done
> show policy map myMappingPolicy
1)      Name: myMappingPolicy
        Source Domain: www.mycompany.com        Source Url:
        Target Domain: www.myrealcompany.com    Target Url:
 Done
>
```

Following is an example of mapping a URL suffix to a different URL suffix:

```
> add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.myrealcompany.com -su /news
 Done
> show policy map myOtherMappingPolicy
1)      Name: myOtherMappingPolicy
        Source Domain: www.mycompany.com        Source Url: /news.html
        Target Domain: www.myrealcompany.com    Target Url: /realnews.html
 Done
>
```

Parameters for creating a mapping policy

mapPolicyName

The name of the map policy you are creating.

sd

The publicly known source domain name. This is the domain name with which a client request arrives at a reverse proxy virtual server for cache redirection.

su

The source URL. Specify all or part of the source URL, in the following format: `/ [[prefix] [*]] [.suffix]`

td

The domain name sent to the server. The source domain name is replaced with this name.

tu

The target URL. Specify the target URL in the following format: `/ [[prefix] [*]] [.suffix]`

To configure a mapping policy for reverse proxy mode by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Map**.
2. In the details pane, click **Add**.
3. In the **Create Map Policy** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a mapping policy" as shown:
   - **Name***- mapPolicyName
   - **Source Domain***-sd
   - **Target Domain***-td
   - **Source URL**-su
   - **Target URL**-tu

   * A required parameter
4. Click **Create**, and then click **Close**. The **Map** pane displays the new mapping policy.

To bind the mapping policy to the cache redirection virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind the mapping policy to the cache redirection virtual server and verify the configuration:

- bind cr vserver <name> **-policyName** <string> [<targetVserver>]
- show cr vserver <name>

Example

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-CR
 Done
> show cr vserver Vserver-CRD-3
      Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
      State: UP
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default: Vserver-LB-CR  Content Precedence: RULE       Cache: REVERSE
```

```
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Policy:         Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)      Map: myMappingPolicy Target: Vserver-LB-CR
 Done
>
```

To bind the mapping policy to the cache redirection virtual server by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server from which you want to bind the mapping policy, and then click **Open**.
3. In the **Configure Virtual Server(Cache Redirection)**, on the **Policies** tab, select **Map**, and then click **Insert Policy**.
4. In the **Policy Name** column, select the policy from drop down list.
5. In the **Target** column, click the down arrow, and then select the vserver from drop down list.
6. Click **OK**.

Selective Cache Redirection

Selective cache redirection sends requests for particular types of content, for example, images, to one cache server or group of cache servers and sends other types of content to a different cache server or group of cache servers. You can configure advanced cache redirection in transparent, reverse proxy, or forward proxy modes.

In selective cache redirection, the NetScaler appliance intercepts a client request and forwards non-cacheable requests to the original destination in the client request. For cacheable requests, the appliance sends the requests to the destination cache server that can serve content of a specific content type.

Selective cache redirection involves configuring content switching policies in addition to cache redirection policies. The NetScaler first evaluates the cache redirection policies that are bound to the cache redirection virtual server. If a request matches a cache redirection policy, the cache redirection virtual server sends the request to the origin server or a load balancing virtual server for the origin. If no cache redirection policies match the request, the NetScaler evaluates the content switching policies bound to the cache redirection virtual server. If a content switching policy matches the request, the cache redirection virtual server redirects the request to a load balancing virtual server for the cache.

To configure selective cache redirection, first enable cache redirection, load balancing, and content switching on the NetScaler appliance. Then, configure a load balancing virtual server for the cache and an associated HTTP service. After this, configure a cache redirection virtual server and bind both the cache redirection and content switching policies to it. Once you have bound the policies, you can configure the virtual server to give precedence to either rule based or URL based content-switching policies.

When configured for transparent mode cache redirection in an edge deployment topology, the NetScaler sends all cacheable HTTP traffic to a transparent cache farm. Clients access the Internet through the NetScaler, which is configured as a Layer 4 switch that receives traffic on port 80.

The NetScaler can direct requests for images (for example, .gif and .jpg files) to one server in the transparent cache farm, and all other requests for static content to other servers in the farm. For this configuration, you configure content switching policies to send images to the image cache and send all other cacheable content to a default cache.

Note: The configuration described here is for transparent selective cache redirection. Therefore, it does not requre a load balancing virtual server for the origin, as would a reverse proxy configuration.

To configure this type of selective cache redirection, first enable cache redirection, load balancing, and content switching. Then, configure a load balancing virtual server for the cache and configure an associated HTTP service. Then, configure a cache redirection virtual server and create and bind both cache redirection and content switching policies to this virtual server.

For details on how to enable cache redirection and load balancing on the NetScaler, see Configuring Cache Redirection.

Enabling Content Switching

To configure selective cache redirection, after you enable both the load balancing and cache redirection features on the NetScaler, you must enable content switching.

To enable content switching by using the NetScaler command line

At the NetScaler command prompt, type:

- enable ns feature cs
- show ns feature

Example

```
> enable ns feature cs
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging              WL               ON
 2)   Surge Protection         SP               ON
 3)   Load Balancing           LB               ON
 4)   Content Switching         CS               ON
 5)   Cache Redirection         CR               ON
       ...
       ...
       ...
 23)  HTML Injection           HTMLInjection      ON
 24)  NetScaler Push           push             OFF
Done
```

To enable cache redirection and load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Configure basic features**.
3. In **Configure Basic Features** dialog box, select the check box next to the **Content Switching**, and then click **OK**.
4. In **Enable/Disable Feature(s)?** dialog box, click **Yes**.

Configuring a Load Balancing Virtual Server for the Cache

Create a load balancing virtual server and an HTTP service for each type of cache server that will be used. For example, if you want to serve JPEG files from one cache server and GIF files from another cache server, and use a third cache server for the rest of the content, create an HTTP service and virtual server for each of the three types of cache servers. Then bind each service to its respective virtual server.

For details on how to create a load balancing virtual server, see Creating a Virtual Server.

For details on how to configure services that represent the cache server, see Configuring an HTTP Service.

For details on how to bind the service to a virtual server, see Binding/Unbinding a Service to/from a Load Balancing Virtual Server.

For details on how to create a transparent proxy cache redirection server, see Configuring a Cache Redirection Virtual Server, and create a virtual server of type TRANSPARENT.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Binding Policies to the Cache Redirection Virtual Server.

Configuring a Cache Redirection Policy for a Specific Type of Content

To identify requests that contain a .gif or .jpeg extension as cacheable, you configure a cache redirection policy and bind it to the cache redirection virtual server.

Note: If a request matches a policy, the NetScaler appliance forwards it to the origin server. As a result, in the following procedure, you configure policies to match requests that do *not* have ".gif" or ".jpeg" extensions.

To configure cache redirection for a specific type of content, configure a policy that uses a simple expression, as described in Configuring a Cache Redirection Policy.

Configuring Policies for Content Switching

You must create a content switching policy to identify specific types of content to be cached in one cache server or farm and identify other types of content to serve from another cache server or farm. For example, you can configure a policy to determine the location for image files with .gif and .jpeg extensions.

After defining the content switching policy, you bind it to a cache redirection virtual server and specify a load balancing virtual server. Requests that match the policy are forwarded to the named load balancing virtual server. Requests that do not match the content switching policy are forwarded to the default load balancing virtual server for the cache.

For more details about the content switching feature and configuring content switching policies, see Content Switching.

You must first create the content switching policy and then bind it to the cache redirection virtual server.

To create a content switching policy by using the NetScaler command line

At the NetScaler command line, type:

- add cs policy <policyName> [**-url** <string> | **-rule** <expression>]
- show cs policy [<policyName>]

Examples

```
> add cs policy Policy-CS-JPEG -rule "REQ.HTTP.URL == '/*.jpeg'"
 Done
> show cs policy Policy-CS-JPEG
        Rule: REQ.HTTP.URL == '/*.jpeg'         Policy: Policy-CS-JPEG
        Hits: 0
 Done
>

> add cs policy Policy-CS-GIF -rule "REQ.HTTP.URL == '/ *.gif'"
 Done
> show cs policy Policy-CS-GIF
        Rule: REQ.HTTP.URL == '/ *.gif'         Policy: Policy-CS-GIF
        Hits: 0
 Done
>
```

```
> add cs policy Policy-CS-JPEG-URL -url /*.jpg
 Done
> show cs policy Policy-CS-JPEG-URL
        URL: /*.jpg     Policy: Policy-CS-JPEG-URL
        Hits: 0
 Done
>

> add cs policy Policy-CS-GIF-URL -url /*.gif
 Done
> show cs policy Policy-CS-GIF-URL
        URL: /*.gif     Policy: Policy-CS-GIF-URL
        Hits: 0
 Done
>
```

Parameters for creating a content switching policy

policyName

Name of the content switching policy. This is a mandatory parameter, and the value cannot be changed after the policy is created.

url

The URL, with wildcards. Specify the string value in this format: // [[prefix] [*]] [.suffix] Maximum value: 208

rule

An expression that the NetScaler evaluates to identify non-cacheable requests. Can consist of multiple expressions joined by AND and OR operators.

To create a URL-based content switching policy by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create Content Switching Policy** dialog box, in the **Name** text box, type a name for the policy.
4. Select the **URL** radio button.
5. In the **Value** text box, type the string value (for example, **/sports**).
6. Click **Create** and click **Close**. The policy you created appears in the **Content Switching Policies** page.

To create a rule-based content switching policy by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create Content Switching Policy** dialog box, in the **Name** text box, type a name for the policy.
4. Select the **Expression** radio button, and then click **Configure**.
5. In the **Create Expression** dialog box, choose the expression syntax that you want to use.
   - If you want to use default syntax, accept the default and proceed to the next step.
   - If you want to use classic syntax, click **Switch to Classic Syntax**.

   The **Expression** portion of the dialog box changes to match your choice. The default syntax **Expression** view has fewer elements than does the classic syntax **Expression** view. In the default syntax **Expression** view, instead of a preview window, a button provides access to an expression evaluator. The evaluator evaluates the expression you entered, to verify that it is valid, and displays an analysis of the expression's effect.
6. Enter your policy expressions.
   - If you are using classic syntax and need further instructions, see Configuring Classic Policies and Expressions chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.
   - If you are using the default syntax and need further instructions, see Configuring Default Syntax Expressions: Getting Started and the chapters describing various types of default syntax expressions.
7. Click **Create** and click **Close**. The policy you created appears in the **Content Switching Policies** pane.

To bind the content switching policy to a cache redirection virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind the content switching policy to a cache redirection virtual server and verify the configuration:

- bind cs vserver <name> <targetVserver> [**-policyName** <string>]
- show cs vserver [<name>]

Example

```
> bind cs vserver Vserver-CR-1 lbcachejpeg -policyName Policy-CS-JPEG
 Done
> bind cs vserver Vserver-CR-1 lbcachegif -policyName Policy-CS-GIF
 Done
> show cs vserver Vserver-CR-1
        Vserver-CR-1 (10.102.29.60:80) - HTTP    Type: CONTENT
        State: UP
        Last state change was at Fri Jul  2 12:53:45 2010
        Time since last state change: 0 days, 00:00:58.920
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Cacheable: YES
        Vserver IP and Port insertion: OFF
        Case Sensitivity: ON
        Push: DISABLED  Push VServer:
        Push Label Rule: none

1)      Policy: Policy-CS-JPEG  Target: lbcachejpeg      Priority: 0     Hits: 0
2)      Policy: Policy-CS-GIF   Target: lbcachegif       Priority: 0     Hits: 0
 Done
>
```

Parameters for binding a content switching policy to a cache redirection virtual server

name

The name of the cache redirection virtual server to which you are binding the content switching policy.

targetVserver

The name of the load balancing virtual server to which cacheable requests that match the content switching policy are sent.

policyName

The name of the content switching policy that decides the cache server to which cacheable content of a specific type is sent.

Note: You are binding the content switching policy to a cache redirection virtual server and not to a content switching virtual server, even though you are using the ' bind cs vserver ' command.

To bind the content switching policy to a cache redirection virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the policy (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Policies** tab, click **CSW**, and then click **Insert Policy**.
4. In the **Policy Name** column, select the policy that you want to configure for the content switching virtual server.
5. In the **Target** column, click the green arrow, and select the target load balancing virtual server from the list.
6. Click **OK**.

Configuring Precedence for Policy Evaluation

You can configure a content switching policy based on either a rule, which is a generic configuration to accommodate various content types, or a URL, which is more specific and defines exactly the type of content that has to be sent to a particular cache server. Essentially, the same content can be defined by either a rule based policy or a URL based policy.

Once you bind content switching policies of either type to a cache redirection virtual server, you can configure the virtual server to give precedence to either rule based or URL based policies. This will, in turn, decide which servers the particular requests are directed to.

To configure precedence for policy evaluation, use the **precedence** parameter, which specifies the type of policy (URL or RULE) that takes precedence on the content redirection virtual server.

Possible values: RULE, URL

Default value: RULE

To configure precedence for policy evaluation by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure precedence for policy evaluation and verify the configuration:

- set cr vserver <name> [**-precedence** (RULE | URL)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
```

```
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
```

To configure precedence for policy evaluation by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure precedence, (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Advanced** tab, next to **Precedence**, click **Rule** or **URL**, and then click **OK**.

Administering a Cache Redirection Virtual Server

To administer a cache redirection virtual server, you need to view cache redirection statistics. You might need to enable or disable cache redirection servers, or direct policy hits to the cache instead of the origin. Administrative tasks also include backing up a cache redirection virtual server and managing client connections.

Viewing Cache Redirection Virtual Server Statistics

You can view properties of a cache redirection virtual server and statistics on the traffic that has passed through a cache redirection virtual server. You can also view the cache redirection virtual servers and policies that you have bound to load balancing virtual servers.

To view statistics for a specific cache redirection virtual servers, use the **name** parameter to specify the name of the virtual server for which statistics will be displayed. Otherwise, statistics for all cache redirection virtual servers are displayed. Maximum Length: 127

To view statistics for a cache redirection virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

stat cr vserver [<name>]

Example

```
> stat cr vserver Vserver-CRD-1

Vserver Summary
                  IP  port    Protocol       State
Vser...CRD-1       0.0.0.0    80        HTTP          UP

VServer Stats:
                              Rate (/s)             Total
Requests                              0                 0
Responses                             0                 0
Request bytes                         0                 0
Response bytes                        0                 0

 Done
>
```

To view statistics for a cache redirection virtual server by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to view statistics, (for example, **Vserver-CRD-1**), and then click **Statistics**.

Omit the server name to display basic statistics for all cache redirection virtual servers. Include the server name to display detailed statistics for that virtual server, including number and size of requests and responses that pass through the virtual server

To view the statistics of a cache redirection virtual server by using the monitoring and dashboard utilities

1. To view the statistics by using the monitoring utilities, click the **Monitoring** tab.
2. In the **Select Group** drop-down menu, choose **CR Virtual Servers**. A list of cache redirection virtual servers appears.
3. To view the statistics by using the dashboard utilities, click the **Dashboard** tab.
4. Click **Applet Client** or **Web Start Client** next to **Statistical Utility**.
5. In the **Select Group** drop-down menu, choose **CR Virtual Servers**. The dashboard displays summary statistics for the cache redirection virtual servers.
6. To see a chart of virtual server activity, click **Chart**. A graphical representation of the virtual server statistics appears.

Enabling or Disabling a Cache Redirection Virtual Server

When you create a cache redirection virtual server, it is enabled by default. If you disable a cache redirection virtual server, its state changes to OUT OF SERVICE and it stops redirecting cacheable client requests. However, the NetScaler appliance continues to respond to ARP and ping requests for the IP address of this virtual server.

To Enable or Disable a cache redirection virtual servers by using the NetScaler command line

At the NetScaler command line, type one of the following commands:

- enable cr vserver <name>
- show cr vserver <name>
- disable cr vserver <name>
- show cr vserver <name>

Examples

```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
       Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
       State: UP  ARP:DISABLED
       Client Idle Timeout: 180 sec
       Down state flush: ENABLED
       Disable Primary Vserver On Down : DISABLED
       Default:        Content Precedence: URL Cache: TRANSPARENT
       On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
       Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
>

> disable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
```

```
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: OUT OF SERVICE  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
```

To Enable or Disable a cache redirection virtual servers by using theconfiguration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to enable or disable, (for example, **Vserver-CRD-1**), and then click **Statistics**.
3. In the **Proceed** dialog box, click **Yes**.

Directing Policy Hits to the Cache Instead of the Origin

By default, when a request matches a policy, the NetScaler appliance forwards the request either to the origin server directly, or to a load balancing virtual server for the origin, depending on how you have configured cache redirection.

You can change the default behavior so that when a request matches a policy, the request is forwarded to a load balancing virtual server for the cache.

To change the destination for a policy hit to the origin or the cache, use the **onPolicyMatch** parameter, which specifies where to send requests that match the cache redirection policy.

The valid options are:

1. CACHE - Directs all matching requests to the cache.
2. ORIGIN - Directs all matching requests to the origin server.

Note: For this option to work, you must select the cacheredirection type as POLICY.

Possible values: CACHE, ORIGIN

Default value: ORIGIN

To change the destination for a policy hit to the origin or the cache by using the NetScaler command line

At the NetScaler command prompt, type the following commands to change the destination for a policy hit and verify the configuration:

- set cr vserver <name> [**-onPolicyMatch** (ORIGIN | CACHE)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
```

```
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
```

To change the destination for a policy hit to the origin or the cache by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to change the destination for a policy hit, (for example, **Vserver-CRD-1**), and then click **Open**.
3. In **Configure Virtual Server (Cache Redirection)** dialog box, click **Advanced** tab.
4. Select **CACHE** or **ORIGIN** from the **Redirect To** drop-down list.
5. Click **OK**.

Backing Up a Cache Redirection Virtual Server

Cache redirection can fail if the primary virtual server fails, or if it is unable to handle excessive traffic. You can specify a backup virtual server to take over the processing of traffic when the primary virtual server fails.

To specify a backup cache redirection virtual server, use the **backupVServer** parameter, which specifies Backup Virtual Server. Maximum Length: 127

To specify a backup cache redirection virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to specify a backup cache redirection virtual server and verify the configuration:

- set cr vserver <name> [**-backupVServer** <string>]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
```

To specify a backup cache redirection virtual server by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to change the destination for a policy hit, (for example, **Vserver-CRD-1**), and then click **Open**.
3. In **Configure Virtual Server (Cache Redirection)** dialog box, select the **Advanced** tab.
4. In the **Backup Virtual Server** drop-down list, select the virtual server.
5. Click **OK**.

Managing Client Connections for a Virtual Server

You can configure timeouts on a cache redirection virtual server so that client connections are not kept open indefinitely. You can also insert Via headers in requests. To possibly reduce network congestion, you can reuse open TCP connections. You can enable or disable delayed cleanup of cache redirection virtual server connections.

You can configure the NetScaler to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.

The following settings can be made on a virtual server:

- When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
- When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

Configuring Client Timeout

You can specify expiration of client requests by setting a timeout value for the cache redirection virtual server. The timeout value is the number of seconds for which the cache redirection virtual server waits to receive a response for the client request.

To configure a time-out value, use the **cltTimeout** parameter, which specifies the time, in seconds, after which the NetScaler appliance closes any idle client connections. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

To configure client timeout by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure client timeout and verify the configuration:

- set cr vserver <name> [**-cltTimeout** <secs>]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
```

To configure client timeout by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click **Open**.
3. In **Configure Virtual Server (Cache Redirection)** dialog box, select the **Advanced** tab.
4. In the **Client Time-out(secs)** text box, enter the time-out value in seconds.
5. Click **OK**.

Inserting Via Headers in the Requests

A Via header lists the protocols and recipients between the start and end points for a request or a response and informs the server of proxies through which the request was sent. You can configure the cache redirection virtual server to insert a Via header in each HTTP request. The **via** parameter is enabled by default when you create a cache redirection virtual server.

To enable or disable Via-header insertion in client requests, use the **via** parameter, which specifies the state of the system in inserting a Via header in the HTTP requests.

Possible values: ON, OFF

Default value: ON

To enable or disable Via-header insertion in client requests by using the NetScaler command line

At the NetScaler command prompt, type:

- set cr vserver <name> [**-via** (ON|OFF)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 6000 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
      Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
>
```

To enable or disable Via-header insertion in client requests by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click **Open**.
3. In **Configure Virtual Server (Cache Redirection)** dialog box, select the **Advanced** tab.
4. Select the **Via** check box.
5. Click **OK**.

Reusing TCP Connections

You can configure the NetScaler appliance to reuse TCP connections to the cache and origin servers across client connections. This can improve performance by saving the time required to establish a session between the server and the NetScaler. The reuse option is enabled by default when you create a cache redirection virtual server.

To enable or disable the reuse of TCP connections, use the **reuse** parameter, which specifies the state of reuse of TCP connections to the cache or origin servers across client connections.

Possible values: ON, OFF

Default value: ON

To enable or disable the reuse of TCP connections by using the NetScaler command line

At the NetScaler command prompt, type:

- set cr vserver <name> [-reuse (ON|OFF)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 6000 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
      Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
```

To enable or disable the reuse of TCP connections by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click **Open**.
3. In **Configure Virtual Server (Cache Redirection)** dialog box, select the **Advanced** tab.
4. Select the **Reuse** check box.
5. Click **OK**.

Configuring Delayed Connection Cleanup

The down state flush option performs delayed cleanup of connections on a cache redirection virtual server. The down state flush option is enabled by default when you create a cache redirection virtual server.

To enable or disable the down state flush option, set the **downStateFlush** parameter.

Possible values: ENABLED, DISABLED

Default value: ENABLED

To enable of disable the down state flush option by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure delayed connection clean up and verify the configuration:

- set cr vserver <name> [**-downStateFlush** (ENABLED | DISABLED)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
```

To enable or disable the reuse of TCP connections by using the configuration utility

1. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click **Open**.
3. In **Configure Virtual Server (Cache Redirection)** dialog box, click **Advanced** tab.
4. Select the **Down state flush** check box.
5. Click **OK**.

N-Tier Cache Redirection

To efficiently handle large amounts of cached data, typically several gigabytes per second, an Internet Service Provider (ISP) deploys several dedicated cache servers. The cache redirection feature of the NetScaler appliance can help load balance the cache servers, but a single appliance or a couple of appliances might not efficiently handle the large volume of traffic.

You can solve the problem by deploying the NetScaler appliances in two tiers (layers), where the appliances in the upper tier load balance those in the lower tier and the appliances in the lower tier load balance the cache servers. This arrangement is called *n-tier cache redirection*.

For purposes such as auditing and security, an ISP has to track client details such as the IP address, information provided, and the time of the interaction. Therefore, client connections through a NetScaler appliance have to be fully transparent. However, if you configure transparent cache redirection, with the NetScaler appliances deployed in parallel, the IP address of the client has to be shared among all the appliances. Sharing of the client IP address creates a conflict that makes network devices, such as routers, cache servers, origin servers, and other NetScaler appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

How N-tier Cache Redirection Is Implemented

To solve the problem, NetScaler n-tier cache redirection splits the source port range among the appliances in the lower tier and includes the client IP address in the request sent to the cache servers. The upper-tier NetScaler appliances are configured to do sessionless load balancing in order to avoid unnecessary load on the appliances.

When the lower-tier NetScaler appliance communicates with a cache server, it uses a mapped IP address (MIP) to represent the source IP address. Therefore, the cache server can identify the NetScaler from which it received the request and send the response to the same NetScaler.

The lower-tier NetScaler appliance inserts the client IP address into the header of the request sent to the cache server. The client IP in the header helps the NetScaler to determine the client to which the packet should be forwarded when it receives the response from a cache server, or the origin server in case of a cache miss. The origin server determines the response to be sent according to the client IP inserted in the request header.

The origin server sends the response to an upper-tier NetScaler, including the source port number from which the origin server received the request. The entire source port range, 1024 to 65535, is distributed among the lower-tier NetScaler appliances. Each lower-tier appliance is exclusively assigned a group of addresses within the range. This allotment enables the upper-tier appliance to unambiguously identify the lower-tier NetScaler appliance that sent the request to the origin server. The upper-tier appliance can therefore forward the response to the correct lower-tier appliance.

The upper-tier NetScaler appliances are configured to do policy-based routing, and the routing policies are defined to determine the IP address of the destination NetScaler from the source port range.

Setup Necessary for Configuring N-Tier CRD

The following setup is necessary for the functioning of n-tier cache redirection:

For each upper-tier NetScaler appliance:

- Enable Layer 3 mode.
- Define policies for policy-based routes (PBRs) so that traffic is forwarded according to the range of the destination port.
- Configure a load balancing virtual server.
- Configure the virtual server to listen to all the traffic coming from the client. Set the Service Type/Protocol to be ANY and IP Address as asterisk (*).
- Enable sessionless load balancing with MAC-based redirection mode to avoid unnecessary load on the upper-tier NetScaler appliances.
- Make sure that the Use Proxy Port option is enabled.
- Create a service for each lower-tier NetScaler and bind all the services to the virtual server.

For each lower-tier NetScaler appliance,

- Configure the cache redirection port range on the NetScaler. Assign an exclusive range to each lower-tier NetScaler.
- Configure a load balancing virtual server and enable MAC-based redirection.
- Create a service for each cache server that is to be load balanced by this NetScaler. When creating the service, enable insertion of client IP in the header. Then, bind all the services to the load balancing virtual server.
- Configure a transparent mode cache redirection virtual server with the following settings:
    - Enable the Origin USIP option.
    - Add a source IP expression to include the client IP in the header.
    - Enable the Use Port Range option.

How N-Tier Cache Redirection Works During a Cache Hit

The following figure shows how cache redirection works when a client request is cacheable and the response is sent from a cache server.

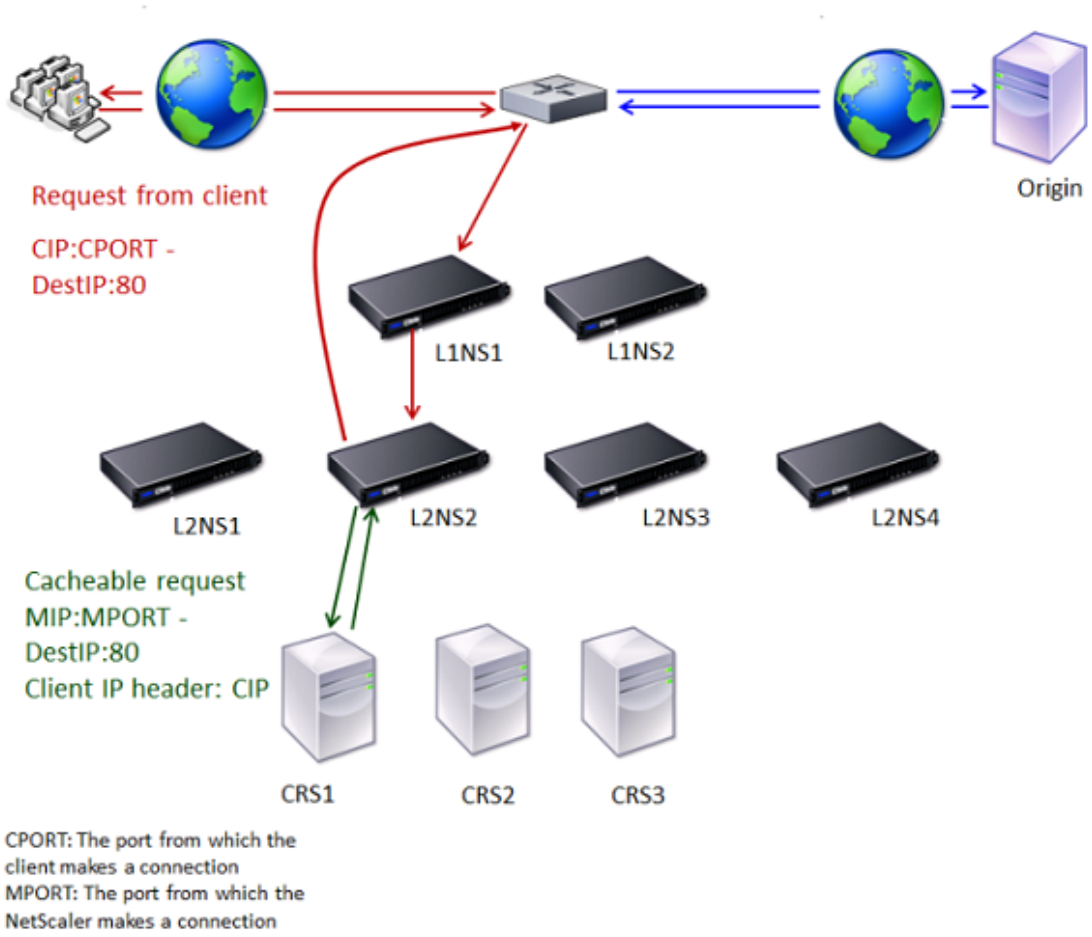


Figure 1. Cache Redirection in Case of a Cache Hit

Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic Flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1, and the request is cacheable. L2NS2 includes the client IP in the request header.
4. CRS1 sends the response to L2NS2 because L2NS2 used its MIP as the source IP address when connecting to CRS1.
5. With the help of the client IP address in the request header, L2NS2 identifies the client from which the request came. L2NS2 directly sends the response to the router, avoiding unnecessary load on the NetScaler in the upper tier.
6. The router forwards the response to Client A.

How N-Tier Cache Redirection Works During a Cache Bypass

The following figure shows how cache redirection works when a client request is sent to an origin server for a response.

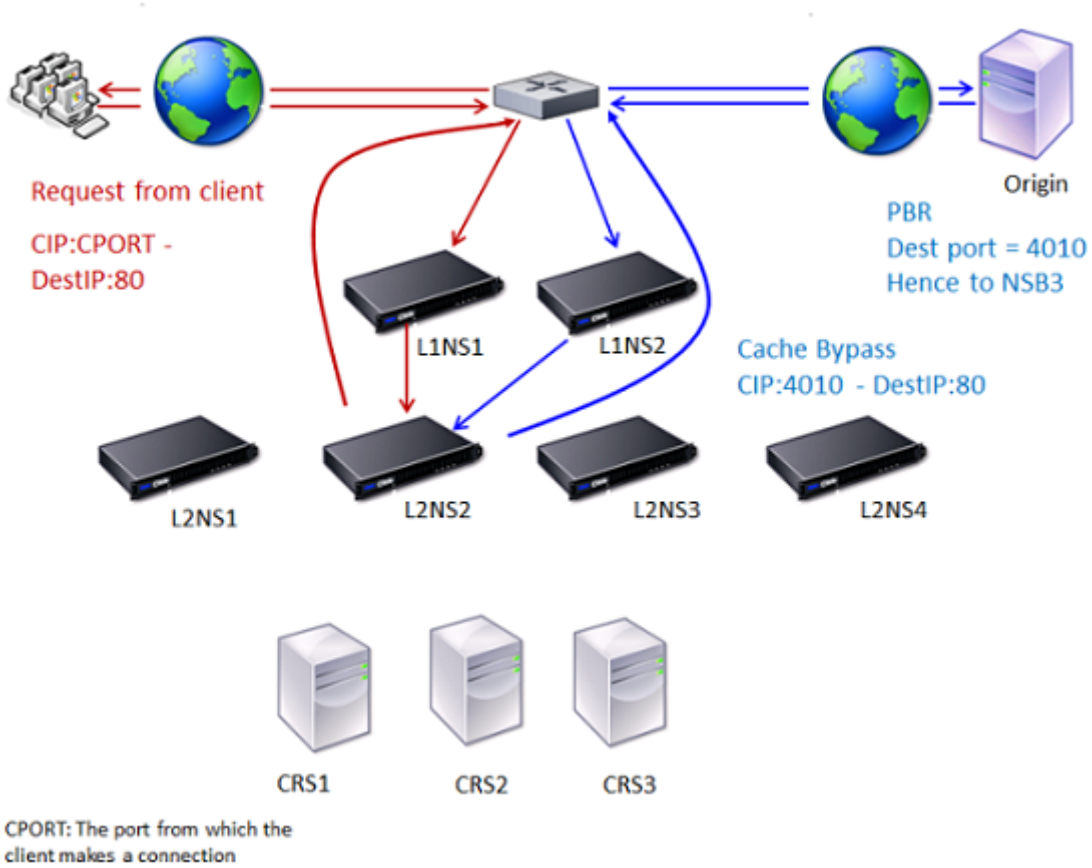


Figure 2. Cache Redirection in Case of a Cache Bypass

Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic Flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. The request is uncacheable (cache bypass). Therefore, L2NS2 sends the request to the origin server through the router.
4. The origin server sends the response to an upper-tier NetScaler, L1NS2.
5. According to the PBR policies, L1NS2 forwards the traffic to the appropriate NetScaler in the lower tier, L2NS2.
6. L2NS2 uses the client IP address in the request header to identify the client from which the request came and sends the response directly to the router, avoiding unnecessary load on the NetScaler in the upper tier.
7. The router forwards the response to Client A.

How N-Tier Cache Redirection Works During a Cache Miss

The following figure shows how cache redirection works when a client request is not cached.

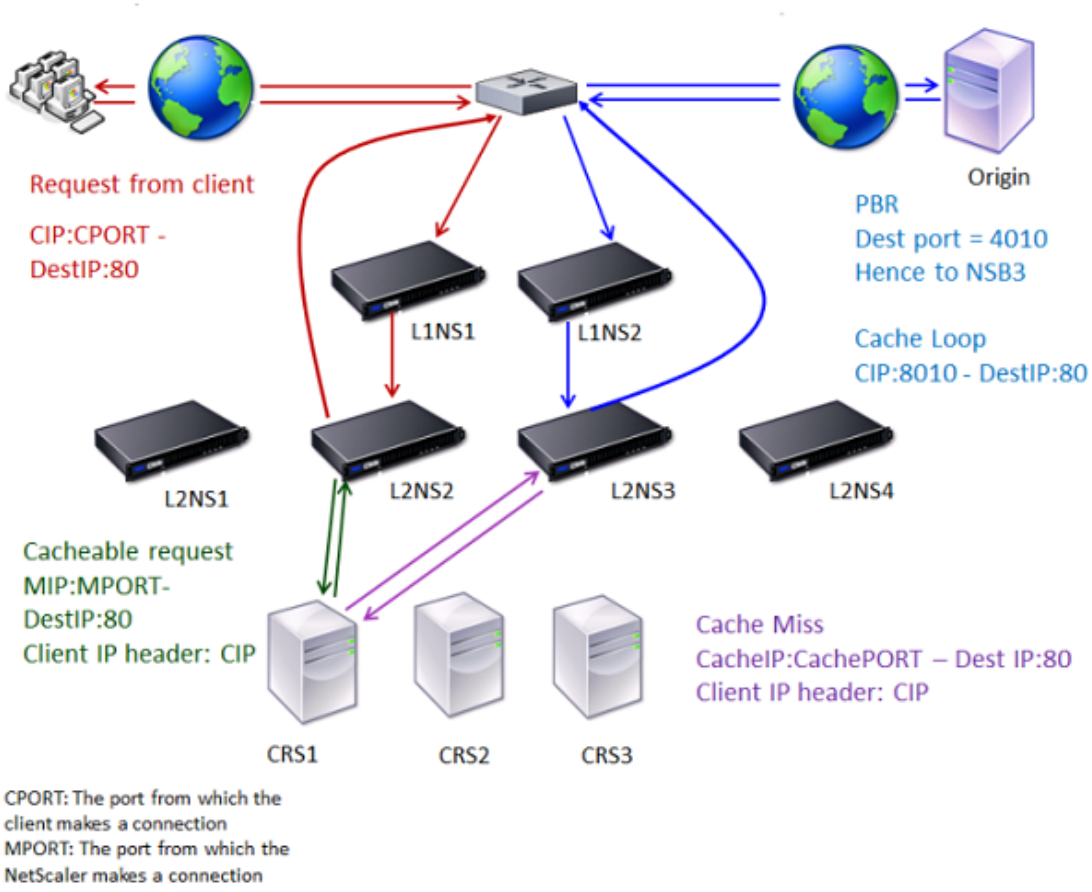


Figure 3. Cache Redirection in Case of a Cache Miss

Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic Flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1 because the request is cacheable.
4. CRS1 does not have the response (cache miss). CRS1 forwards the request to the origin server through the NetScaler in the lower tier. L2NS3 intercepts the traffic.
5. L2NS3 takes the client IP from the header and forwards the request to the origin server. The source port included in the packet is the L2NS3 port from which the request is sent to the origin server.
6. The origin server sends the response to an upper-tier NetScaler, L1NS2.
7. According to the PBR policies, L1NS2 forwards the traffic to the appropriate NetScaler in the lower tier, L2NS3.
8. L2NS3 forwards the response to the router.
9. The router forwards the response to Client A.

Configuring the Upper-Tier NetScaler Appliances

Configure each of the upper-tier NetScaler appliances as follows.

To configure an upper-tier appliance for n-tier cache redirection by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add service <serviceName> <serviceIP> <serviceType> <port>

  Run this command for each service to be added.

- add lb vserver <vServerName> ANY * <port> **-persistenceType** <persistenceMethod> **-lbMethod** <lbMethod> **-m** MAC **-sessionless** ENABLED **-cltTimeout** <client_Timeout_Value>

- bind lb vserver <vServerName> <serviceName>

  Run this command for each service to be bound.

- enable ns mode l3

- add ns pbr <pbrName> <action> **-srcPort** <sourcePortNumber> **-destPort** <startPortNumber-endPortNumber> **-nexthop** <serviceIpAddress> **-protocol** TCP

- apply pbrs

  Run this command after adding all the necessary PBRs.

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format.

When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, RDP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure an upper-tier appliance for n-tier cache redirection by using the configuration utility

1. Enable L3 mode:
    a. In the navigation pane, click **System**, and then click **Settings**.
    b. In the **Settings** group, click the **Configure modes** link.
    c. Select the **Layer 3 Mode (IP Forwarding)** check box.
    d. Click **OK**.
2. Configure policy-based routing (PBR):
    a. In the navigation pane, click **Network**, and then click **PBRs**.
    b. In the **Policy-Based Routing (PBRs)** pane, click **Add**.
    c. Type a name for the PBR.
    d. Select the action as **Allow**.
    e. In the **Next Hop** box, type the IP address of the service, which represents a lower-tier NetScaler.
    f. Select **TCP** from the **Protocol** drop-down list.
    g. Type the source port and the range of the destination port corresponding to the lower-tier NetScaler being added.
    h. Click **Create**.
    i. In the details pane, select the **PBR** and click **Apply**.
    j. Repeat Step (i) to Step (vii) for each lower-tier NetScaler.
3. Create a service for each lower-tier NetScaler:
    a. In the navigation pane, expand **Load Balancing**, and then click **Services**.
    b. In the details pane, click **Add**.
    c. Specify the name, protocol, IP address, and port. The protocol should be **ANY**.
    d. Click **Create**.
4. Configure a load balancing virtual server:
    a. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
    b. In the details pane, click **Add**.
    c. Specify the name, protocol, IP address, and port. The protocol should be **ANY** and the IP address should be *****.

d. In the **Services** tab, select the services that represent the lower-tier NetScaler appliances.

e. In the **Advanced** tab, select the **Redirection Mode** as **MAC Based** and select the **Sessionless** check box.

f. Click **Create**.

Configuring the Lower-Tier NetScaler Appliances

Configure each of the lower-tier NetScaler appliances as follows.

To configure a lower-tier appliance for n-tier cache redirection by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add service <cacheServiceName> <cacheServiceIP> <serviceType> <port> **-cip** ENABLED "ClientIP" **-cachetype** transparent

  Repeat for each cache server.

- add lb vserver <cachevServerName> <serviceType> **-m** MAC

- bind lb vserver <cachevServerName> <cacheServiceName>

  Repeat for each cache server.

- add cr vserver <vServerName> <serviceType> ***** <port> **-srcIPExpr** "HTTP.REQ.HEADER(\"ClientIP\")" **-originusip** ON **-cacheVserver** <cachevServerName> **-usePortRange** ON

- set ns config **-crPortRange** <startPortNumber-endPortNumber>

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with

this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, RDP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure a lower-tier appliance for n-tier cache redirection by using the configuration utility

1. Create a service for each cache server. To create a service:
    a. In the navigation pane, expand **Load Balancing**, and then click **Services**.
    b. In the details pane, click **Add**, and specify the name and protocol. Clear the **Directly Addressable** check box.
    c. In the **Advanced** tab, select the **Override Global** check box and the **Client IP** check box, and then in the **Header** box, type **ClientIP**.
    d. In the **Cache Type** box, select **Transparent Cache**.
    e. Click **Create**.
2. Configure a load balancing virtual server:
    a. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
    b. In the details pane, click **Add** and specify the name, protocol, IP address, and port. The IP address should be an asterisk (*).
    c. In the **Services** tab, select the services that represent the cache servers.
    d. In the **Advanced** tab, for **Redirection Mode**, select **MAC Based**.
    e. Click **Create**.
3. Configure a cache redirection virtual server:
    a. In the navigation pane, expand **Cache Redirection**, and then click **Virtual Servers**.
    b. In the details pane, click **Add** and specify the name, protocol, IP address, and port. The IP address should be *****.
    c. For **Cache Type**, select **Transparent**.
    d. On the **Advanced** tab, in the **Cache Server** box, select the new load balancing virtual server and check the **Origin USIP** and **Use Port Range** check boxes. In the **Source IP Expression** box, type `HTTP.REQ.HEADER("ClientIP")`.
    e. Click **Create**.
4. Assign a source port range for the NetScaler:
    a. In the navigation pane, click **System**, and then click **Settings**.
    b. In the **Settings** group, click the **Change global system settings** link.
    c. In the **Cache Redirection Port Range** group, specify the port range for the NetScaler by typing a port number for **Start Port** and a port number for **End Port**.
    d. Click **OK**.

Client Keep-Alive

The Citrix® NetScaler® appliance provides features such as client keep-alive to improve the performance of a transaction management environment. Performance of any transaction management environment depends on factors such as bandwidth usage, download time, speed of the server and the client networks, and time consumed to complete a transaction. Client keep-alive improves performance by reducing the time consumed in a transaction. Two other settings that affect connection management include the maximum number of HTTP connections retained in the connection reuse pool and whether or not connection multiplexing is enabled. These variables are configured with HTTP profiles.

Client keep-alive enables multiple client requests to be sent on a single client connection. This feature helps in a transaction management environment where the server closes the client connection after serving the response to the client. The client has to open a new connection for each request to the server leading to a lot of time being consumed for a transaction. The client keep-alive feature of the appliance alleviates this problem by keeping the client-side connection open between the client and the appliance even after the server closes the client connection. This allows multiple client requests to be sent using a single connection. Keeping the client-side connection open saves the packet round trips associated with opening and closing a connection. This is most beneficial to SSL sessions because this eliminates unnecessary termination and open sequences, thus reducing the time taken for a transaction to occur.

Client keep-alive is useful under either of the following conditions:

- When the server does not support client keep-alive.
- When the server supports client keep-alive but an application on the server does not support client keep-alive.

Client keep-alive can be applied to all HTTP services globally, or to a specific service, such as HTTP or SSL. Note that client keep-alive applies only to HTTP and HTTPS services.

The following figure illustrates a typical client keep-alive deployment.

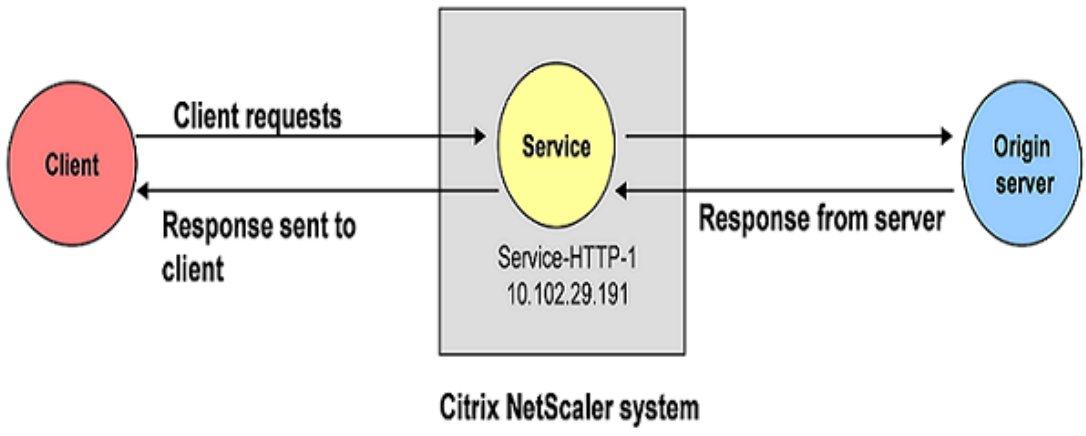


Figure 1. Client Keep-Alive Entity Model

As shown in the figure, to configure client keep-alive, you need to define services and enable client keep-alive for those services. Services represent applications on physical servers. The traffic from the client is intercepted by the configured service and the client request is directed to the origin server. The server sends the response and closes the connection between the server and the appliance. If a “Connection: Close” header is sent, this header is corrupted in the client-side response, and the client-side connection is kept open. As a result, the client does not have to open a new connection for the next request; instead, the connection to the server is reopened.

Note: If a server sends back two “Connection: Close” headers, only one is edited. This results in significant delays on the client rendering of the object because a client does not assume that the object has been delivered completely until the connection is actually closed.

Configuring Client Keep-Alive

To configure the client keep-alive feature, you need to create a service and enable client keep-alive for that service.

The service you configure enables the NetScaler appliance to keep the client-side connection open even after the server has closed the connection between the server and the appliance.

Note that while configuring a service, if the client keep-alive option is not explicitly specified, the service uses the global setting for the client keep-alive connection. For more information, see Enabling or Disabling Client Keep-Alive Globally.

To create a service with client keep-alive enabled by using the NetScaler command line

At the NetScaler command prompt, type:

add service (<name> | <IPaddress>) <serviceType> <port> **-CKA** (YES | NO)

Example

add service Service-HTTP-1 10.102.29.191 HTTP 80 -CKA YES

Parameters for creating a service with client keep-alive enabled

name

The name of the service. The service name must not exceed 31 characters.

IPaddress

The IP address of the origin server, which the service represents.

port

The port on which the service listens. The port number must be a positive number not greater than 65535. The minimum value is 1.

serviceType

The type of service that is being added. This parameter determines the behavior of the service. The possible values are: HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP.

CKA

The state of the client keep-alive feature for the service. Possible values: YES, NO.

To create a service with client keep-alive enabled by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service with client keep-alive enabled” as shown:
   - **Service Name***—name (Cannot be changed for an existing service)
   - **Server***—IPaddress
   - **Port***—port
   - **Protocol***—serviceType
   - **Client Keep-Alive**—name (See **Advanced** tab and select the **Override Global** check box.)

   *A required parameter
4. Click **Create**, and then click **Close**.

Configuring Client Keep-Alive

To configure the client keep-alive feature, you need to create a service and enable client keep-alive for that service.

The service you configure enables the NetScaler appliance to keep the client-side connection open even after the server has closed the connection between the server and the appliance.

Note that while configuring a service, if the client keep-alive option is not explicitly specified, the service uses the global setting for the client keep-alive connection. For more information, see Enabling or Disabling Client Keep-Alive Globally.

To create a service with client keep-alive enabled by using the NetScaler command line

At the NetScaler command prompt, type:

add service (<name> | <IPaddress>) <serviceType> <port> **-CKA** (YES | NO)

Example

add service Service-HTTP-1 10.102.29.191 HTTP 80 -CKA YES

Parameters for creating a service with client keep-alive enabled

name

The name of the service. The service name must not exceed 31 characters.

IPaddress

The IP address of the origin server, which the service represents.

port

The port on which the service listens. The port number must be a positive number not greater than 65535. The minimum value is 1.

serviceType

The type of service that is being added. This parameter determines the behavior of the service. The possible values are: HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP.

CKA

The state of the client keep-alive feature for the service. Possible values: YES, NO.

To create a service with client keep-alive enabled by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service with client keep-alive enabled” as shown:
   - **Service Name***—name (Cannot be changed for an existing service)
   - **Server***—IPaddress
   - **Port***—port
   - **Protocol***—serviceType
   - **Client Keep-Alive**—name (See **Advanced** tab and select the **Override Global** check box.)

   *A required parameter
4. Click **Create**, and then click **Close**.

Enabling or Disabling Client Keep-Alive Globally

The client keep-alive feature is disabled on the system by default. If you enable client keep-alive globally, all new services inherit the global settings by default. The following procedure describes the steps to enable or disable client keep-alive globally.

To enable or disable the client keep-alive mode globally by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable mode cka
- disable mode cka

To enable or disable the client keep-alive mode globally by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Configure modes**.
3. In the **Configure Modes** dialog box, select or clear the **Client Keep-Alive** check box.
4. Click **OK**.

Enabling or Disabling Client Keep-Alive for a Service

You can enable or disable client keep-alive at the service level. Note that the service level settings take precedence over the global settings. The following procedure describes the steps to enable or disable client keep-alive at the service level.

To enable or disable the client keep-alive mode for a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service (<name> | <IPaddress>) **-CKA** (YES | NO)

Example

set service Service-HTTP-1 -CKA YES

To enable or disable the client keep-alive mode for a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click the service for which you want to enable or disable client keep-alive, and then click **Open**.
3. In the **Configure Service** dialog box, on the **Advanced** tab, under **Settings**, select or clear the **Client Keep-Alive** check box.
4. Click **OK**.

Configuring Connection Options with HTTP Profiles

An HTTP profile is a collection of configuration settings that is used to control HTTP requests to and responses from virtual servers on the NetScaler appliance. You can use HTTP profiles to configure the maximum number of HTTP connections retained in the connection reuse pool. You can also use HTTP profiles to enable connection multiplexing if you are not sure that it is already enabled. (Connection multiplexing is enabled by default.) For more information about HTTP profiles, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

To configure the number of connections in the reuse pool by using the NetScaler command line

At the NetScaler command prompt, type:

set ns httpprofile <profileName> **-maxReusePool** <value>

To configure the number of connections in the reuse pool by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Profiles**.
2. In the details pane, on the **HTTP Profiles** tab, select the HTTP profile for which you want to configure the number of connections, and then click **Open**.
3. In the **Configure HTTP Profiles** dialog box, in **Max connections in reusepool**, specify the maximum number of connections you want to allow in connection reuse pool.
4. Click **OK**.

To enable connection multiplexing by using the NetScaler command line

At the NetScaler command prompt, type:

set ns httpprofile <profileName> **-conMultiplex** ENABLED

To enable connection multiplexing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Profiles**.
2. In the details pane, on the **HTTP Profiles** tab, select the HTTP profile for which you want to enable connection multiplexing, and then click **Open**.
3. In the **Configure HTTP Profiles** dialog box, select the **Connection Multiplexing** check box. Note that this check box is enabled by default.
4. Click **OK**.

Compression

The Citrix® NetScaler® appliance compression feature compresses the size of HTTP responses sent from servers to compression-aware browsers and thereby improves the performance of Web sites by reducing the download time of Web content. Bandwidth is also saved in the process. Another indirect benefit of HTTP compression is that the data passed between the Web server and the browser is encrypted by virtue of the compression algorithm, adding more security to the data.

After you enable the compression feature and compression is set at the service level, global compression policies are enabled, and the NetScaler can compress data for traffic that matches these policies. You can augment the built-in compression policies by creating new compression actions and policies and binding the policies globally or to particular virtual servers.

When you bind a policy globally, you specify a bind point, which corresponds to a step in the sequence in which traffic is processed. Policies bound to a virtual server have their own place in the sequence. Therefore, binding the policies affects the order in which they are evaluated. Alternatively, you can associate policies with policy labels that are not associated with a bind point. Such policies can be invoked only by other policies.

You can view statistics for compressed data that the NetScaler transmits.

Note: The compression feature uses both classic and advanced policies. This content is best understood if you are familiar with configuration principles for basic policies, virtual servers, and services. For more information about policies, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX123868http://support.citrix.com/article/CTX128673. For more information about virtual servers and services, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869http://support.citrix.com/article/CTX128670.

The NetScaler can compress HTML and other content that is generated statically or dynamically, including MIME types such as text/html, text/plain, text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint.

After you enable the compression feature on the NetScaler and set compression ON for HTTP and SSL services, built-in compression policies are applied to the HTTP and SSL services. You can disable compression on a particular service, and you can create custom compression policies and bind both the built-in and custom compression policies to a load balancing virtual server.

Enabling or Disabling Compression

If you want the Citrix® NetScaler® appliance to compress data, enable the compression feature and set compression ON for the configured service. After enabling compression, the built-in compression policies are in effect, and compression is automatically enabled for any services that you create.

If you want to use compression in a load balancing environment, you also enable the load balancing feature. To compress traffic that is sent over SSL, you also enable the SSL feature.

To enable or disable compression, load balancing, and SSL by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable feature cmp lb ssl
- disable feature cmp lb ssl

To enable or disable compression, load balancing, and SSL by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Configure basic features**.
3. In the **Configure Basic Features** dialog box, select the **Compression** check box to enable it; clear the check box to disable it. If appropriate, also select the **Load Balancing** and **SSL Offloading** check boxes.
4. Click **OK**, and click **Yes** in the **Enable/Disable Feature(s)?** message box.

Enabling or Disabling Compression

If you want the Citrix® NetScaler® appliance to compress data, enable the compression feature and set compression ON for the configured service. After enabling compression, the built-in compression policies are in effect, and compression is automatically enabled for any services that you create.

If you want to use compression in a load balancing environment, you also enable the load balancing feature. To compress traffic that is sent over SSL, you also enable the SSL feature.

To enable or disable compression, load balancing, and SSL by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable feature cmp lb ssl
- disable feature cmp lb ssl

To enable or disable compression, load balancing, and SSL by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Configure basic features**.
3. In the **Configure Basic Features** dialog box, select the **Compression** check box to enable it; clear the check box to disable it. If appropriate, also select the **Load Balancing** and **SSL Offloading** check boxes.
4. Click **OK**, and click **Yes** in the **Enable/Disable Feature(s)?** message box.

Enabling and Disabling Compression for a Service

By default, if the compression feature is disabled, any new service that you create is disabled for compression and uses the built-in compression policies. If you create any services prior to enabling compression, you must manually enable compression for the service.

You can disable or enable compression for HTTP and SSL services.

Compression is in effect for a compression-enabled service once you bind the service to a vserver. You can bind HTTP services to an HTTP load balancing vserver, and bind SSL services to an SSL load balancing vserver. The protocol types must match. For additional information about configuring HTTP and SSL-based virtual servers, see the *Citrix NetScaler Traffic Management Guide* at .

To create a compression-enabled service by using the NetScaler command line

At the NetScaler command prompt, type:

- add service <name> <IPAddress> **HTTP** <portNumber>
- set service <name> **-CMP** YES

Example

```
add service Service-HTTP-1 10.102.29.51 HTTP 80
set service Service-HTTP-1 -CMP YES
```

Parameters for a compression-enabled service

name

The name of the service. This is a required parameter and cannot be changed. The service name must not exceed 127 characters.

IPAddress

The IP address of the origin server, which the service represents.

port

The port on which the service listens. The port number must be a positive number not greater than 65535. The minimum value is 1.

serviceType

The type of service that is being added. This parameter determines the behavior of the service. Possible values: HTTP and SSL.

cmp

State of the HTTP compression feature for the service. Possible values: YES and NO. Default: NO.

To enable or disable service-level compression by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, on the **Services** tab, select the service for which you want to enable or disable compression, and then click **Open**.
3. In the **Configure Service** dialog box, on the **Advanced** tab, under **Settings**, select **Override Global**, and then select or clear the **Compression** check box.
4. Click **OK**, and then click **Close**.

Configuring Compression Actions

You associate actions with compression policies. If an HTTP request matches the policy rule, the action is applied to the response. For example, you can configure a compression policy that identifies requests that are sent to a particular server, and associate the policy with an action that compresses data that is sent with the response.

There are four built-in compression actions:

- **COMPRESS:** Uses the GZIP algorithm to compress data for browsers that support either GZIP or both GZIP and DEFLATE. The NetScaler appliance uses the DEFLATE algorithm to compress data for browsers that support the DEFLATE algorithm. If the browser does not support either algorithm, and the action has been set to COMPRESS, the NetScaler appliance does not compress data.
- **NOCOMPRESS:** Does not compress data.
- **GZIP:** Uses the GZIP algorithm to compress data for browsers that support GZIP compression. If the browser does not support the GZIP algorithm the NetScaler appliance does not compress data.
- **DEFLATE:** Uses the DEFLATE algorithm to compress data for browsers that support the DEFLATE algorithm. If the browser does not support the DEFLATE algorithm, and the action has been set to DEFLATE, the NetScaler appliance does not compress data.

Compression actions determine whether and what type of compression the NetScaler appliance applies to a response. After creating an action, you associate the action with one or more compression policies.

To create a compression action by using the NetScaler command line

At the NetScaler command prompt, type:

add cmp action <name> **-cmpType** (compress|gzip|deflate|nocompress)

Parameters for compression

name

The name of the compression action. This is a required parameter and cannot be changed.

cmpType

The type of compression action. Possible values: compress, gzip, deflate, nocompress.

To create a compression action by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Compression Action** dialog box, in the **Name** text box, type the name of the action (for example, Action-CMP-1).
4. Under **Compression Type**, choose the compression type (for example, GZIP).
5. Click **Create**, and then click **Close**.

If you try to delete a built-in action, or any action that is associated with a policy, an error message appears. Only custom actions that have no associated policy can be deleted.

To delete a compression action by using the NetScaler command line

At the NetScaler command prompt, type:

rm cmp action <actionName>

To delete a compression action by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Actions**.
2. In the details pane, select the compression action that you want to delete.
3. Click **Remove**, and then click **Yes** in the **Remove** message box.

Configuring Compression Policies

A compression policy contains a rule, which is a logical expression that enables the NetScaler appliance to identify the traffic that should be compressed.

When the Citrix® NetScaler® appliance receives an HTTP response from a server, it evaluates a built-in or custom compression policy to determine whether to compress the response and the type of compression to apply.

There are five built-in classic and advanced compression policies. These policies are activated globally when you enable compression.

The following table describes the built-in compression policies.

Table 1. Built-in Classic and Advanced Policies for Compression

Built-in Classic and Advanced Compression Policies	Description
ns_nocmp_mozilla_47 ns_adv_nocmp_mozilla_47	Does not compress CSS files when a request is sent from a Mozilla 4.7 Web browser.
ns_cmp_mscss ns_adv_cmp_mscss	Compresses CSS files when the request is sent from a Microsoft Internet Explorer Web browser.
ns_cmp_msapp ns_adv_cmp_msapp	Compresses files that are generated by the following applications: • Microsoft Office Word • Microsoft Office Excel • Microsoft Office PowerPoint
ns_cmp_content_type ns_adv_cmp_content_type	Compresses data when the response contains the header 'Content-Type' and contains text.
ns_nocmp_xml_ie ns_adv_nocmp_xml_ie	Does not compress when a request is sent from a Microsoft Internet Explorer browser with the response header 'Content-Type' and contains text or xml.

To view built-in compression policies by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policies**.
2. In the details pane, view the built-in compression policies.

You can create a compression policy by using the built-in compression actions and named expressions, or you can use custom actions and expressions.

To create a compression policy by using the NetScaler command line

At the NetScaler command prompt, type:

add cmp policy <name> **-resAction** (compress|gzip|deflate|nocompress) **-rule** <build_in_rule_name>|“<user_defined_rule>”

Example

add cmp policy Policy-CMP-2 -resAction gzip -rule HTTP.REQ.HEADER(“User-Agent”).CONTAINS(“Mozilla/4.7”

Parameters for compression policies

name

The name of the compression policy. This is a required parameter and the value cannot be changed. Maximum length: 127 characters.

resAction

The type of compression action that is performed when a response matches the policy. Possible values: compress, gzip, deflate, nocompress.

rule

The rule that the NetScaler appliance uses to determine whether to compress an HTTP response.

To create a compression policy by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create Compression Policy** dialog box, in the **Policy Name** text box, type the name of the policy.
4. In **Response Action**, do one of the following:
   - To use a built-in or existing compression action, choose a compression action in the drop-down list.
   - To create a new compression action, click **New**. In the **Create Compression Action** dialog box, enter a compression action name and type, and then click **Create**.
5. In the **Expression** box, either type the default syntax expression or do one of the following:
   - Click **Prefix**, and then select the expression prefix you want. Then, select the flow type (**REQ** or **RES**), and then enter a period (.) to display a list of functions that can be used with the flow type. Type a period after each function until you have entered the expression you want. Click the **Operators** button to insert an operator (for example, a Boolean or relational operator).
   - Click **Add**, and then select a named expression.
6. If you want to evaluate your expression, click **Evaluate**.
7. If you want to clear the **Expression** box, click **Clear**.
8. Click **Create**, and then click **Close**.

You can modify the actions and expressions that are associated with a user-defined policy. However, you cannot make any modifications to the built-in compression policies.

To modify a compression policy by using the NetScaler command line

At the NetScaler command prompt, type:

set cmp policy <policyName> **-resAction** (compress|gzip|deflate|nocompress) **-rule** <rule>

Example

set cmp policy Policy-CMP-2 -resAction NOCOMPRESS -rule HTTP.REQ.URL.SUFFIX.EQ("html")

For the equivalent classic syntax example, see the To modify a compression policy by using the NetScaler command line in Classic Policy Examples for Compression.

To modify a compression policy by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policies**.
2. In the details pane, click the policy you want to modify, and then click **Open**.
3. In the **Configure Compression Policy** dialog box, select the action or expression, and then modify it.
4. Click **OK**.

You can remove a compression policy if the policy is not bound globally or to a vserver. If the compression policy is bound, you must first unbind it. You cannot remove a built-in compression policy.

To remove an unbound compression policy by using the NetScaler command line

At the NetScaler command prompt, type:

rm cmp **-policyName** <policyName>

To remove an unbound compression policy by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policies**.
2. In the details pane, click the policy you want to remove, and then click **Remove**.

Bind Points and Order of Evaluation for Default Syntax Compression Policies

For a default syntax policy to take effect, you must ensure that the policy is invoked at some point during the Citrix® NetScaler® appliance's processing of traffic. To specify the invocation time, you associate the policy with a bind point. The following are the built-in bind points, listed in order of evaluation:

- **Request-time override.** If a request matches a request-time override policy, by default request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time load balancing virtual server.** If policy evaluation cannot be completed after all the request-time override policies are evaluated, the NetScaler appliance processes request-time policies that are bound to load balancing virtual servers. If the request matches one of these policies, evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time content switching virtual server.** Policies that are bound to this bind point are evaluated after request-time policies that are bound to load balancing virtual servers.
- **Request-time default.** If policy evaluation cannot be completed after all request-time, virtual server-specific policies are evaluated, the NetScaler appliance processes request-time default policies. If the request matches a request-time default policy, by default request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Response-time override.** Similar to request-time override policy evaluation.
- **Response-time load balancing virtual server.** Similar to request-time virtual server policy evaluation.
- **Response-time content switching virtual server.** Similar to request-time virtual server policy evaluation.
- **Response-time default.** Similar to request-time default policy evaluation.

You can associate multiple policies with each bind point. To control the order of evaluation of the policies associated with the bind point you configure a priority level. In the absence of any other flow control information, policies are evaluated according to priority level, starting with the lowest numeric priority value.

You can also bind a given default syntax compression policy to more than one bind point. However, the bind point at which the policy is evaluated first is determined by the order specified in the preceding list. Assigned priority values must be unique within the collection of policies bound to a specific bind point. Additionally, for a given policy, only one binding is allowed per bind point at any given time. You should bind a policy with an INVAL action to a request-time override or a response-time override bind point. To delete a policy, you must first unbind it.

Table 1. Entries to Control Evaluation Flow in a Policy Label

Attribute	Specifies
Name	The name of the policy that you bound to the virtual server.
Bound To	The name of the virtual server to which the policy is bound.
Priority	The priority level used to determine when the policy is evaluated relative to other policies that are bound to this virtual server. Specify an integer. The lower the integer, the higher the priority.
Goto Expression	Determines the next policy to evaluate in this label. Goto can proceed only forward in a policy label. Omitting the Goto expression is the same as specifying END. You can provide one of the following values: • **NEXT:** Go to the policy with the next higher priority. • **END:** Stop evaluation. • **USE_INVOCATION_RESULT:** Applicable if this entry invokes another policy label. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy label performs a NEXT. • **Positive number:** Priority number of the next policy to be evaluated. • **Numeric expression:** Expression that produces the priority number of the next policy to be evaluated.
Flow Type	You must specify a flow type to determine whether this policy is evaluated at request time or response time.
Invoke Label Type	Designates a policy label type. The value can be one of the following: • **Request Vserver:** Invokes request-time policies that are associated with a virtual server. • **Response Vserver:** Invokes response-time policies that are associated with a virtual server. • **Policy label:** Invokes another policy label, as identified by the policy label for the bank.
Invoke Label Name	The name of a virtual server or a policy label, depending on the value that you specified for the invocation type.

Finally, you can also bind compression policies to custom bind points—bind points that you create. The bind points that you create are called *policy labels*. For more information about policy labels, see Creating Policy Labels.

Bind Points and Order of Evaluation for Classic Compression Policies

You can bind a classic compression policy either at the global level or at the virtual server (content switching or load balancing virtual server) level. You can also bind a given compression policy to more than one bind point. The priority values that are assigned to a policy that is bound to various bind points need not have the same value. For example, if you bind a policy called *mycompressionpolicy* both globally and to a load balancing virtual server, you can assign the policy a priority of 10 at the global bind point and a priority of 100 at the virtual server bind point. If you do not assign a policy a priority value, the NetScaler appliance assigns the classic compression policy a default priority value of 0 (zero).

Therefore, the priorities assigned to the classic compression policies that are bound to various bind points at any given time might be an assortment of custom priority values and default values. During classic policy evaluation, priority values are considered first. The NetScaler appliance begins with the policy that has the lowest priority value (the default value of 0, if any, or the next higher custom priority value if none of the policies have a priority value of 0) regardless of the bind point to which the policy is bound. Then, the appliance evaluates policies in ascending order of priority values while moving from one bind point to the other if necessary.

When the NetScaler appliance is evaluating policies based on their priority values, if multiple policies bound to different bind points happen to have the same priority value, the bind points of those policies are considered, with the order of evaluation progressing from the most specific bind point to the least specific bind point. For example, if classic policies are bound globally, to a content switching virtual server, and to a load balancing virtual server, the policies bound to the load balancing virtual server are evaluated first because, in this configuration, the load balancing virtual server represents the most specific bind point. The NetScaler appliance then evaluates the policies that are bound to the content switching virtual server. Finally, the appliance evaluates the policies that are bound at the global level (which is the least specific bind point in this configuration).

Finally, at a given bind point, if two or more policies have the same priority value, the chronological order in which the policies were configured is considered. Among the policies that are bound to the same bind point, with the same priority value, the policy that was configured first is evaluated first, followed by the policy that was configured second, and so on. In this way, the NetScaler appliance evaluates classic policies based first on assigned priorities, then on bind points, and finally on the chronological order in which the policies were configured.

Creating Policy Labels

You can create compression policy labels and configure banks of policies for these new labels.

Policy labels can be considered as abstract bind points that you create. You can create policy labels only for advanced policies. The policies that are bound to a policy label can be evaluated by invoking the policy label from a policy that is bound to one of the following bind points:

- Request-time override
- Request-time default
- Response-time override
- Response-time default
- A virtual server

You can invoke a policy label any number of times, unlike a policy which can only be invoked once.

To create a policy label for compression by using the NetScaler command line

At the NetScaler command line, type:

add cmp policylabel <labelName> **-evaluates** (REQ | RES)

Parameters for creating compression policy labels

Name

The name of the policy label. This is a required parameter and the value cannot be changed. Maximum length: 127 characters.

Evaluates

Determines when this policy label is applied: request or response time.

Priority

Determines the numeric priority of the policy. Policies with lower priority values are evaluated before policies with higher priority values.

Policy Name

The name of a policy in this policy label.

Expression

The policy rule. Classic and advanced policy expressions are described in detail in the *Citrix NetScaler Policy Configuration and Reference Guide* at .

Action

The action taken when traffic matches this policy.

Goto Expression

Optional. Determines the next policy to evaluate in this bank. Omitting a value is the same as specifying END. Goto can only proceed forward in a policy label. You can specify one of the following values:

- NEXT. Go to the policy with the next higher priority.
- END. Stop evaluation.
- USE_INVOCATION_RESULT. Applicable if this entry invokes another policy label. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy label performs a NEXT.
- Positive number. The priority number of the next policy to be evaluated.
- Numeric expression. An expression that produces the priority number of the next policy to be evaluated.

Invoke

Optional. You can invoke other policy labels by using this option. A policy label can be a set of policies that is bound to a load balancing or content switching virtual server, or it can be a set of policies that is bound to a policy label. In this field, you select the name of a policy label or a virtual server. After evaluating the policies in the invoked bank, the appliance continues evaluating policies that are bound to the current policy label (the selected bind point).

To create a policy label for compression by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policy Labels**.
2. In the details pane, click **Add**.
3. In the **Create Compression Policy Label** dialog box, in **Name**, specify a name for the policy label.
4. In the **Evaluates** drop-down list, select whether the policy label will be evaluated at request time (**REQ**) or response time (**RES**).
5. Click **Insert Policy**, and then select the policy, or click **New Policy** to create a new policy.

   Note: To ensure that the NetScaler appliance processes the policy label at the right time, you can configure an invocation of this label from the policy labels that are associated with the built-in bind points. Select the appropriate policy label of request vserver from the **Invoke** column field.
6. Click **Create**, and then click **Close**.

   Note: You can use the NOPOLICY “dummy” policy to invoke any policy label from another policy label. The NOPOLICY entry is a placeholder that does not process a rule.

Binding Compression Policies Globally

A global compression policy applies to all services that support compression. When binding the policy, you assign it a priority. The policy is enabled by default upon creation.

To globally bind a compression policy by using the NetScaler command line

At the NetScaler command prompt, type:

bind cmp global <policyName> **-priority** <positiveInteger> **-state** (enabled|disabled)

To unbind a globally bound compression policy by using the NetScaler command line

At the NetScaler command prompt, type:

unbind cmp global <policyName>

Entries to control globally bound compression policies

Priority

A numeric value that indicates when this policy is evaluated relative to others. A lower priority is evaluated before a higher one.

Note that in the configuration utility, you can click the Priority field and edit the priority level or drag the entry to a new position in the policy label. If you drag the entry to a new position, the priority level is updated automatically.

Policy name

The name of the policy.

Expression

The rule that the NetScaler appliance uses to determine whether to compress an HTTP response. For details, see To create a compression policy by using the configuration utility.

Action

The action that is performed when traffic matches this policy.

State

Indicates whether the globally bound policy is enabled or disabled.

Insert Policy

Adds a new policy to this policy label.

Note that in the configuration utility, when you click this option, a drop-down list appears in the **Policy Name** field. You can select from the following:

- **Policy name.** The name of an existing policy.
- **New policy.** Invokes the policy creation editor.

Request

A type of bind point. At request time, policies that are bound to this bind point are evaluated.

Response

A type of bind point. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.

Override Global

A type of bind point. Binds policies globally, which makes them available to all virtual servers. When a request flows through a feature, the appliance first evaluates global request-time override policies. At response time, the appliance starts with policies that are bound to the global response-time override bind point.

Default Global

Binds policies to the default bind point. At request time, if policy evaluation cannot be completed after all request-time policies for virtual servers have been evaluated, the appliance processes request-time default policies. At response time, if policy evaluation cannot be completed after all response-time policies for virtual servers have been evaluated, the appliance processes response-time default policies.

Goto Expression

Optional. Determines the next policy to evaluate in this bank. You can provide one of several values. NEXT means go to the policy with the next higher priority. END means stop evaluation. USE_INVOCATION_RESULT is applicable if this entry invokes another policy label. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy label performs a NEXT. You can also enter a positive number that equals the priority number of the next policy to be evaluated. Finally, you can enter a numeric expression that produces the priority number of the next policy to be evaluated. The Goto can only proceed forward in a policy label. If you omit the Goto expression, it is the same as specifying END.

Invoke

Optional. In this field, you select the name of a policy label or a virtual server. A policy label is a set of policies that is bound to a load balancing or content switching virtual server. After evaluating the policies in the invoked policy label, the appliance continues evaluating policies that are bound to the current policy label (the selected bind point).

To globally bind a compression policy by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **Bind/Unbind Compression Policies to Global** dialog box, do one of the following:
   - To bind compression policies by using classic expressions, click **Classic Expression**. Click **Insert Policy**, and then click the policy name that you want to bind. Double-click the **Priority** field for the policy and set the priority. A lower value causes the policy to be evaluated before policies with a higher priority value. Click **Apply Changes**.

     Optionally, to configure an expression as described in Binding Compression Policies Globally, double-click the field in the **Expression** column, and specify a valid expression.
   - To bind compression policies by using advanced expressions, click **Advanced Policies**. Select a **Request** or **Response** bind point, and then select a second level of binding of either **Override Global** or **Default Global**. A list of policies appears. These are policies that are bound to this bind point. Click **Insert Policy**, and then click the policy name that you want to bind. Double-click the **Priority** field for the policy and set the priority. A lower value causes the policy to be evaluated before policies with a higher priority value. Specify other optional values, such as a Goto expression or invocation of an external policy label.

     To configure a Goto expression, double-click the field in the **Goto Expression** column, and enter valid priority level, the keywords NEXT or END, or an advanced expression.
4. Click **Apply Changes**, and then click **Close**.

To unbind a globally bound compression policy by using the configuration utility

1. In the navigation pane, expand **HTTP Compression**, and then click **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **Bind/Unbind Compression Policies to Global** dialog box, click the policies that you want to unbind, and then click **Unbind Policy**.

Binding Compression Policies to Virtual Servers

When you configure virtual server-based compression, you bind services and compression policies to a virtual server. This causes traffic that flows through a virtual server (to and from the bound services) to be subject to compression policies that you bind to the vserver.

If you bind a policy to a vserver, the policy is evaluated only by compression-enabled services that are bound to this vserver. When binding a policy, you set a priority value. Policies with a lower priority value are evaluated before policies with a higher value. After unbinding a policy from a vserver, the policy ceases to act on the services associated with that vserver.

To bind a compression policy to a load balancing vserver by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <vserverName> **-policyName** <policyName> **-priority** <positiveInt>

To unbind a compression policy from a load balancing vserver by using the NetScaler command line

At the NetScaler command prompt, type:

unbind lb vserver <vserverName> **-policyName** <policyName>

To bind a compression policy to a load balancing vserver by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click the name of the virtual server, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Policies** tab, click **Compression**.
4. Click either **Classic Expression** or **Advanced Expression**. Click **Insert Policy** and select the policy that you want to bind. Optionally, you can double-click the **Priority** field and type a new priority level. To invoke another policy label or to configure a policy for a request vserver, from the **Invoke** drop-down list, make an appropriate selection. If you select a request vserver, you can bind a compression policy for the selected vserver by double-clicking the Invoke field. The **Configure Compression Policies** dialog box appears.
5. Click **OK**, and then click **Close**.

To unbind a compression policy from a load balancing vserver by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click the name of the virtual server for which you want to unbind a compression policy, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Policies** tab, click the name of the policy that you want to unbind, and then click **Unbind Policy**.

Setting Global Compression Parameters

You can customize the way the Citrix® NetScaler® appliance compresses data.

To set global compression parameters by using the NetScaler command line

At the NetScaler command prompt, type:

set cmp parameter [**-cmpLevel** <compressionLevel>] [**-quantumSize** <integer>] [**-serverCmp** (ON | OFF)] [**-minResSize** <positiveInteger>] [**-cmpBypassPct** <positiveInteger>] [**-cmpOnPush** (ENABLED | DISABLED)] [**-policyType** (CLASSIC | ADVANCED)]

Parameters for setting compression

cmpLevel

The compression level can be set to one of the following:

- **Optimal.** Corresponds to a gzip level of 5-7.
- **Best speed.** Corresponds to a gzip level of 1.
- **Best compression.** Corresponds to a gzip level of 9.

The default value is Optimal. Typically, there is little enough practical difference in these settings to warrant changing from the default.

quantumSize

This is the minimum quantum of data to be filled, after which data begins to compress. The minimum value is 1 and the maximum is 63488. Default value: 57344.

serverCmp

This is disabled by default to allow the NetScaler appliance to handle all compression. When enabled, this option allows the server to send compressed data to the NetScaler appliance.

minResSize

The smallest response size to be compressed.

cmpBypassPct

The NetScaler CPU threshold after which compression is not performed.

policyType

The policy type to be implemented, as follows:

- **Classic.** Classic policies evaluate basic characteristics of traffic and other data.
- **Advanced.** Advanced policies can perform the same type of evaluations as classic policies. In addition, advanced policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header).

heurExpiry

This is disabled by default. When enabled, the NetScaler appliance performs heuristic base file expiration.

heurExpiryThres

The threshold compression ratio for heuristic base file expiration, multiplied by one hundred. For example, if you want a threshold ratio of 1.25, specify 125 in the field. The default threshold ratio is 100, and the range is from 1 to 1000.

heurExpiryHistWt

A weight-to-delta-compression history for the heuristic base file expiration, as a percentage. The default weightage value is 50, and the range is 1 to 100.

cmpOnPush

This is disabled by default. When enabled, the NetScaler appliance does not wait for the quantum to be filled before starting to compress the data. Upon receipt of a packet with a PUSH flag, compression of the accumulated packets starts immediately.

To set global compression parameters by using the configuration utility

1. In the navigation pane, click **HTTP Compression**.
2. In the details pane, click **Change compression settings**.
3. In the **Configure Compression Parameters** dialog box, configure the settings (for example, set the Quantum size and Compression level), and then click **OK**.

Configuring Compression for a Load Balancing Virtual Server

When you configure virtual server-based compression, you bind services and compression policies to a virtual server. This causes traffic that flows through a virtual server to and from the bound services to be subject to compression policies that you bind to the vserver.

When a client request flows through a vserver, compression policies identify whether the client can accept compressed data. The Citrix® NetScaler® appliance forwards the request to the destination server, as identified by a service that is bound to the load balancing vserver. After the NetScaler appliance receives the response from the server, it determines whether the response is compressible based on the compression policies that are bound to the virtual server. If the content is compressible, it is compressed and forwarded to the client.

Task overview: configuring compression for a load balancing vserver.

1. Enable compression and load balancing, as described in Enabling and Disabling Compression.
2. Add a vserver, as described in the chapters on load balancing and SSL offloading in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.
3. Add one or more HTTP or SSL services and bind the services to a vserver, as described in the chapters on load balancing and SSL offloading in the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.
4. Create compression policies, as described in Configuring Compression Policies.
5. Bind the compression policies to the vserver, as described in Binding Compression Policies to Virtual Servers.

Viewing Compression Statistics by Using the Dashboard

The Dashboard utility displays summary and detailed compression statistics in tabular and graphic format.

Note: For more information about the statistics and charts, see the Dashboard help on the Citrix® NetScaler® appliance.

To view compression statistics by using the Dashboard

1. In the Dashboard utility, in the **Select Group** list, choose **Compression**, and then do one or more of the following:
   - To view of summary of compression statistics, click the **Summary** tab.
   - To view compression statistics by protocol type, click the **Details** tab.
   - To view the rate of requests processed by the compression feature, click the **Chart** tab.

Viewing Compression Statistics by Using SNMP

You can collect compression statistics in an SNMP monitor. You can view the following compression statistics by using the SNMP network management application.

Note: For more information about SNMP, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

- Number of compression requests (OID: 1.3.6.1.4.1.5951.4.1.1.50.1)
- Number of compressed bytes transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.2)
- Number of compressible bytes received (OID: 1.3.6.1.4.1.5951.4.1.1.50.3)
- Number of compressible packets transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.4)
- Number of compressible packets received (OID: 1.3.6.1.4.1.5951.4.1.1.50.5)
- Ratio of compressible data received and compressed data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.6)
- Ratio of total data received to total data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.7)

Viewing Additional Compression Statistics

When the Citrix® NetScaler® compresses a response based on a policy, the policy hit counter is incremented. You can view statistics for a compression policy, including the number of hits.

To view details and hits to a compression policy by using the command line

At the NetScaler command prompt, type:

sh cmp policy <policyName>

To view a summary of compression statistics by using the command line

At the NetScaler command prompt, type:

stat cmp

To view detailed statistics of compression by using the command line

At the NetScaler command prompt, type:

stat cmp **-detail**

To view compression statistics by using the configuration utility

1. In the navigation pane, click **HTTP Compression**.
2. In the details pane, click the **Statistics** link.

   Note: To view statistics for an individual policy, expand **HTTP Compression**, and then click **Policies**. In the details pane, click the policy for which you want to view statistics.

Content Filtering

Content filtering can do some of the same tasks as the Citrix Application Firewall, and is a less CPU-intensive tool. It is limited, however, to examining the header portion of the HTTP request or response and to performing a few simple actions on connections that match. If you have a complex Web site that makes extensive use of scripts and accesses back-end databases, the Application Firewall may be the better tool for protecting that Web site. For more information about the Citrix Application Firewall, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX128677.

Content filtering is based on regular expressions that you can apply to either HTTP requests or HTTP responses. To block requests from a particular site, for example, you could use an expression that compares each request's URL to the URL specified in the expression. The expression is part of a policy, which also specifies an action to be performed on requests or responses that match the expression. For example, an action might drop a request or reset the connection.

Following are some examples of things you can do with content filtering policies:

- Prevent users from accessing certain parts of your Web sites unless they are connecting from authorized locations.
- Prevent inappropriate HTTP headers from being sent to your Web server, possibly breaching security.
- Redirect specified requests to a different server or service.

To configure content filtering, once you have made sure that the feature is enabled, you configure filtering actions for your servers to perform on selected connections (unless the predefined actions are adequate for your purposes). Then you can configure policies to apply the actions to selected connections. Your policies can use predefined expressions, or you can create your own. To activate the policies you configured, you bind them either globally or to specific virtual servers.

Enabling Content Filtering

By default, content filtering is enabled on NetScaler appliances running the NetScaler operating system 8.0 or above. If you are upgrading an existing appliance from an operating system version earlier than 8.0, you must update the licenses before you can use content filtering, and you may need to enable the content filtering feature itself manually.

To enable content filtering by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable content filtering and verify the configuration:

- enable ns feature ContentFiltering
- show ns feature

Example

```
> enable ns feature ContentFiltering
 Done
> show ns feature

      Feature                 Acronym          Status
      -------                 -------          ------
1)    Web Logging               WL               ON
2)    Surge Protection          SP               OFF
.
.
.
.
11)   Http DoS Protection       HDOSP            OFF
12)   Content Filtering         CF               ON
.
.
23)   HTML Injection            HTMLInjection    ON
24)   NetScaler Push            push             OFF
Done
```

To enable content by filtering by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Content Filtering** check box, and then click **OK**.
4. In the **Enable/Disable feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature is enabled.

Enabling Content Filtering

By default, content filtering is enabled on NetScaler appliances running the NetScaler operating system 8.0 or above. If you are upgrading an existing appliance from an operating system version earlier than 8.0, you must update the licenses before you can use content filtering, and you may need to enable the content filtering feature itself manually.

To enable content filtering by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable content filtering and verify the configuration:

- enable ns feature ContentFiltering
- show ns feature

Example

```
> enable ns feature ContentFiltering
 Done
> show ns feature

      Feature                 Acronym          Status
      -------                 -------          ------
1)    Web Logging               WL               ON
2)    Surge Protection          SP               OFF
.
.
.
.
11)   Http DoS Protection       HDOSP            OFF
12)   Content Filtering         CF               ON
.
.
23)   HTML Injection            HTMLInjection    ON
24)   NetScaler Push            push             OFF
Done
```

To enable content by filtering by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Content Filtering** check box, and then click **OK**.
4. In the **Enable/Disable feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature is enabled.

Configuring a Content Filtering Action

After you enable the content filtering feature, you create one or more actions to tell your NetScaler appliance how to handle the connections it receives.

Content filtering supports the following actions for HTTP requests:

Add

Adds the specified HTTP header before sending the request to the Web server.

Reset

Terminates the connection, sending the appropriate termination notice to the user's browser.

Forward

Redirects the request to the designated service.

Drop

Silently deletes the request, without sending a response to the user's browser.

Corrupt

Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the request to the server.

Content filtering supports the following actions for HTTP responses:

Add

Adds the specified HTTP header before sending the response to the user's browser.

ErrorCode

Returns the designated HTTP error code to the user's browser.

Corrupt

Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the response to the user's browser.

To configure a content filtering action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a Content Filtering action and verify the configuration:

- add filter action <name> <qualifier> [<serviceName>] [<value>] [<respCode>] [<page>]
- show filter action <name>

Example

```
> add filter action act_drop Drop
 Done
> show filter action act_drop
1)      Name: act_drop   Filter Type: drop
 Done
```

Parameters for configuring a content filtering action

name

A name for the filtering action. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the type of action. (Cannot be changed after the action has been created.)

qualifier

The action that you want your content filtering Action to perform. (Cannot be changed after the action has been created.)

servicename

If the Qualifier is Forward, the name of the service to which you want to forward requests. If the Qualifier is Forward, you must configure either a servicename or a page, but not both. Otherwise, you should not set this value.

value

If the Qualifier is Add, the header that you want added. This argument is optional.

respcode

If the Qualifier is ErrorCode, the numeric code you want returned to the user (such as 404, the standard HTTP code for a non-existent Web page). This argument is optional.

page

If the Qualifier is Forward, you must configure either a servicename or a page, but not both. Otherwise, you should not set this value.

To configure a content filtering action by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then select **Filter**.
2. In the details pane, do one of the following:
   - To create a new action, click **Add**.
   - To modify an existing action, select the action, and then click **Open**.
3. In the **Add Filter Action** or **Configure Filter Action** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring a content filtering action" as follows (asterisk indicates a required parameter):
   - Action Name*—name
   - Qualifier*—qualifier (Determines which of the following parameters you can configure)
   - Service Name—servicename
   - HeaderName:Value—value
   - Response Code—respcode
   - Response Page—page
4. Fill in any other required information. For example, if you are configuring an action to send an HTTP error code, you must choose the appropriate error code from a drop-down list. If necessary, you can then modify the text of the error message, which is displayed beneath the drop-down list.
5. Click **Create** or **OK**, and then click **Close**. The **Actions** list displays the action you configured, and a message in the status bar indicates that your action has been created.

Configuring a Content Filtering Policy

To implement content filtering, you must configure at least one policy to tell your NetScaler appliance how to distinguish the connections you want to filter. You must first have configured at least one filtering action, because when you configure a policy, you associate it with an action.

Content filtering policies examine a combination of one or more of the following elements to select requests or responses for filtering:

URL

The URL in the HTTP request.

URL query

Only the query portion of the URL, which is the portion after the query (?) symbol.

URL token

Only the tokens in the URL, if any, which are the parts that begin with an ampersand (&) and consist of the token name, followed by an equals sign (=), followed by the token value.

HTTP method

The HTTP method used in the request, which is usually GET or POST, but can be any of the eight defined HTTP methods.

HTTP version

The HTTP version in the request, which is usually HTTP 1.1.

Standard HTTP header

Any of the standard HTTP headers defined in the HTTP 1.1 specification.

Standard HTTP header value

The value portion of the HTTP header, which is the portion after the colon and space (:).

Custom HTTP header

A non-standard HTTP header issued by your Web site or that appears in a user request.

Custom header value

The value portion of the custom HTTP header, which (as with the standard HTTP header) is the portion after the colon and space (:).

Client Source IP

The IP from which the client request was sent.

Content filtering policies use the simpler of two NetScaler expressions languages, called classic expressions. For a complete description of classic expressions, how they work, and how to configure them manually, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

To configure a content filtering policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a content filtering policy and verify the configuration:

- add filter policy <name> **-rule** <expression> (**-reqAction** <action> | **-resAction** <string>
- show filter policy <name>

Example

```
> add filter policy cf-pol -rule "REQ.HTTP.URL CONTAINS http://abc.com" -reqaction DROP
 Done
> show filter policy cf-pol
1)      Name: cf-pol    Rule: REQ.HTTP.URL CONTAINS http://abc.com
        Request action: DROP
        Response action:
        Hits: 0
 Done
```

Parameters for configuring a content filtering policy

name

A name for the filtering action. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the type of action. (Cannot be changed after the action has been created.)

rule

A NetScaler classic expression that describes the connections you want to select using this policy.

reqaction or resaction

Type -reqaction if your policy applies to requests, or -resaction if your policy applies to responses. You must choose one or the other, but not both.

action

The name of the content filtering action you want to perform on connections that match this policy.

To configure a content filtering policy by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then select **Filter**.
2. In the details pane, to create a new policy, click **Add**.
3. If you are creating a new policy, in the **Create Filter Policy** dialog box, in the **Filter Name** text box, type a name for your new policy.
4. Select either **Request Action** or **Response Action** to activate the drop-down list to the right of that item.
5. Click the down arrow to the right of the drop-down list and select the action to be performed on the request or response. The default choices are **RESET** and **DROP**. Any other actions you have created will also appear in this list.

   Note: You can also click **New** to create a new Content Filtering action, or **Modify** to modify an existing Content Filtering action. You can only modify actions you created; the default actions are read-only.
6. If you want to use a predefined expression (or named expression) to define your policy, choose one from the **Named Expressions** list.

   a. Click the down arrow to the right of the first **Named Expressions** drop-down list, and choose the category of named expressions that contains the named expression you want to use.

   b. Click the down arrow to the right of the second **Named Expressions** drop-down list, and choose the named expression you want. As you choose a named expression, the regular expression definition of that named expression appears in the **Preview Expression** pane beneath the **Named Expression** list boxes.

   c. Click **Add Expression** to add that named expression to the **Expression** list.

   Note: You should perform either this step or step 7, but not both.
7. If you want to create a new expression to define your policy, use the **Expression Editor**.

   a. Click the **Add** button. The **Add Expression** dialog box appears.

   b. In the **Add Expression** dialog box, choose the type of connection you want to filter. The Flow Type is set to **REQ** by default, which tells the NetScaler appliance to look at incoming connections, or requests. If you want to filter outgoing connections (responses), you click the right arrow beside the drop-down list and choose **RES**.

   c. If the Protocol is not already set to HTTP, click the down arrow to the right of the **Protocol** drop-down list and choose HTTP.

      Note: In the NetScaler classic expressions language, “HTTP” includes HTTPS requests, as well.

   d. Click the down arrow to the right of the **Qualifier** drop-down list, and then choose a qualifier for your expression. Your choices are:

METHOD

The HTTP method used in the request.

URL

The contents of the URL header.

URLTOKENS

The URL tokens in the HTTP header.

VERSION

The HTTP version of the connection.

HEADER

The header portion of the HTTP request.

URLLEN

The length of the contents of the URL header.

URLQUERY

The query portion of the contents of the URL header.

URLQUERYLEN

The length of the query portion of the URL header.

The contents of the remaining list boxes change to the choices appropriate to the Qualifier you pick. For example, if you choose **HEADER**, a text field labeled **Header Name*** appears below the **Flow Type** list box.

e. Click the down arrow to the right of the **Operator** drop-down list, and choose an operator for your expression. Your choices will vary depending on the Protocol you chose in the preceding step. The following list includes all of the operators:

==

Matches the following text string exactly.

!=

Does not exactly match the following text string.

>

Is greater than the following integer.

CONTAINS

Contains the following text string.

CONTENTS

The contents of the designated header, URL, or URL query.

EXISTS

The specified header or query exists.

NOTCONTAINS

Does not contain the following text string.

NOTEXISTS

The specified header or query does not exist.

f. If the **Value** text box is visible, type the appropriate string or number. If you are testing a string in any way, type the string into the **Value** text box. If you are testing an integer in any way, type the integer into the **Value** text box.

g. If you chose **HEADER** as the Protocol, type the header you want in the **Header Name*** text box.

h. Click **OK** to add your expression to the **Expressions** list.

i. Repeat steps B through H to create any additional expressions you want for your profile.

j. Click **Close** to close the **Expressions Editor**.

8. If you created a new expression, in the **Expression** frame select an option from the **Match Any Expression** drop-down list. Your choices are:

- **Match Any Expression**. If a request matches any expression in the Expressions list, the request matches this policy.
- **Match All Expressions** If a request matches all expressions in the Expressions list, the request matches this policy. If it does not matchall of them, it does not match this policy.
- **Tabular Expression** Switches the Expressions list to a tabular format with three columns. In the first column you can place a BEGIN [(] operator. The second column contains the expressions you have selected or created. In the third column, you can place any of the other operators in the following list, to create complex policy groups in which each group can be configured for match any expression or match all expressions.
- The AND [&&] operator tells the appliance to require that are quest match both the current expression and the following expression.
- The OR [||] operator tells the appliance to require that arequest match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.
- The END [)] operator tells the appliance that this is the last expression in this expression group or policy.

Note: The Tabular format allows you to create a complex policy that contains both “Match Any Expression” and “Match All Expressions” on a per-expression basis. You are not limited to just one or the other.

- **Advanced Free-Form** Switches off the Expressions Editor entirely and modifies the Expressions list into a text area. In the text area, you can type the PCRE-format regular expression of your choice to define this policy. This is both the most powerful and the most difficult method of creating a policy, and is recommended only for those thoroughly familiar with the NetScaler appliance and PCRE-formatregular expressions.

  Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that is what you want.

9. Repeat steps 6 through 8 to add any additional expressions you want to the **Expressions** list. You can mix named expressions and expressions created in the **Expressions Editor**. To the NetScaler appliance, they are all the same.

10. Click **Create** to create your new policy.Your new policy appears in the **Policies** pane list.

11. Click **Close**. To create additional Content Filtering policies, repeat the previous procedure. To remove a Content Filtering policy, select the policy in the **Policies** tab and click **Remove**.

Binding a Content Filtering Policy

You must bind each content filtering policy to put it into effect. You can bind policies globally or to a particular virtual server. Globally bound policies are evaluated each time traffic directed to any virtual server matches the policy. Policies bound to a specific vserver are evaluated only when that vserver receives traffic that matches the policy.

To bind a policy to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a policy to a virtual server and verify the configuration:

- bind lb vserver <vserverName> **-policyName** <string> **-priority** <positive_integer>
- show lb vserver <vserverName>

Example

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver vs-loadbal
1)      vs-loadbal (10.102.29.20:80) - HTTP      Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

 Done
```

To globally bind a policy by using the NetScaler command line

At a NetScaler command prompt, type the following commands to globally bind a policy and verify the configuration:

- bind filter global (<policyName> [**-priority** <positive_integer>]) [**-state** (ENABLED | DISABLED)]
- show filter global

Example

```
bind filter global cf-pol -priority 1
Done show filter global
1)      Policy Name: cf-pol     Priority: 1
Done
```

To bind a policy to a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind the content filtering policy from the list, and click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, select the **Policies** tab, and then select the check box in the **Active** column of the filter policy that you want to bind to the virtual server.
4. Click **OK**. The policies you have bound display a check mark and the word **Yes** in the **Policies Bound** column of the **Policies** tab.

To globally bind a policy by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then select **Filter**.
2. In the details pane, in the **Policies tab**, select the policy that you want to bind, and then click **Global Bindings**.
3. In the **Bind/Unbind Filter Policies** dialog box, in the **Policy Name** drop-down list, select a policy, and then click **Add**. The policy is added to the **Configured** list.

   Note: To select multiple policies from the list, press and hold the Ctrl key, then click each policy you want.
4. Click **OK**, and then click **Close**. The policies you have bound display a check mark and the word **Yes** in the **Globally Bound** column of the **Policies** tab.

Configuring Content Filtering for a Commonly Used Deployment Scenario

This example provides instructions for using the configuration utility to implement a content filtering policy in which, if a requested URL contains root.exe or cmd.exe, the content filtering policy filter-CF-nimda is evaluated and the connection is reset.

To configure this content filtering policy, you must do the following:

- Enable content filtering
- Configure content filtering policy
- Bind content filtering policy globally or to a virtual server
- Verify the configuration

Note: Since this example uses a default content filtering action, you do not need to create a separate content filtering action.

To enable content filtering

1. In the navigation pane, expand **System**, and click **Settings**.
2. In the details pane, under **Modes & Features**, click **Change Basic Features**.
3. In the **Configure Basic Features** dialog box, select the **Content Filtering** check box, and then click **OK**.
4. In the **Enable/Disable feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature is enabled.

To configure the content filtering policy filter-CF-nimda

1. In the navigation pane, expand **Protection Features**, and click **Filter**.
2. In the details pane, click **Add**. The **Create Filter Policy** dialog box appears.
3. In the **Create Filter Policy** dialog box, in the **Filter Name** text box, type the name `filter-CF-nimda`.
4. Select the **Request Action** option, and in the drop-down list, select **RESET**.
5. In the **Expression frame**, select **Match Any Expression** from the drop-down list, and then click **Add**.
6. In the **Add Expression** dialog box, **Expression Type** drop-down list, select **General**.
7. In the **Flow Type** drop-down list, select **REQ**.
8. In the **Protocol** drop-down list, select **HTTP**.
9. In the **Qualifier** drop-down list, select **URL**.
10. In the **Operator** drop-down list, select **CONTAINS**.
11. In the **Value** text box, type `cmd.exe`, and then click **OK**. The expression is added in the **Expression** text box.
12. To create another expression, repeat Steps 7 through 11, but in the **Value** text box, type `root.exe`. Then click **OK**, and finally click **Close**.
13. Click **Create** on the **Create Filter Policy** dialog box. The filter policy filter-CF-nimda appears in the **Filter** list.
14. Click **Close**.

To globally bind the content filtering policy

1. In the navigation pane, expand **Protection Features**, and select **Filter**. The Filter page appears in the right pane.
2. In the details pane, **Policies** tab, select the policy that you want to bind and click **Global Bindings**. The **Bind/Unbind Filter Policies** dialog box appears.
3. In the **Bind/Unbind Filter Policies** dialog box, in the **Policy Name** drop-down list, select the policy **filter-CF-nimda**, and click **Add**. The policy is added to the **Configured** list.
4. Click **OK**, and then click **Close**. The policy you have bound displays a check mark and **Yes** in the **Globally Bound** column of the **Policies** tab.

To bind the content filtering policy to a virtual server

1. In the navigation pane, expand the **Load Balancing** node and click **Virtual Servers**.
2. In the details pane virtual servers list, select **vserver-CF-1** to which you want to bind the content filtering policy and click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, select the **Policies** tab.
4. In the **Active** column, select the check box for the policy filter-CF-nimda, and then click **OK**. Your content filtering policy is now active, and should be filtering requests. If it is functioning correctly, the **Hits** counter is incremented every time there is a request for a URL containing either root.exe or cmd.exe. This allows you to confirm that your content filtering policy is working. The content filtering policy is bound to the virtual server.

To verify the content filtering configuration by using the NetScaler command line

At the command prompt, type the following command to verify the content filtering configuration:

sh filter policy filter-CF-nimda

Example

```
sh filter policy filter-CF-nimda
      Name: filter-CF-nimda   Rule: REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe
      Request action: RESET
      Response action:
      Hits: 0
Done
```

Note: The **Hits** counter displays an integer that denotes the number of times the `filter-CF-nimda` policy is evaluated. In the preceding steps, the Hits counter is set to zero because no requests for a URL containing either cmd.exe or root.exe have been made yet. If you want to see the counter increment in real time, you can simply request a URL that contains either of these strings.

To verify the content filtering configuration by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then click **Filter**.
2. In the details pane, select the filter policy **filter-CF-nimda**.The bottom of the pane should display the following:

   Request Action:

   RESET

   Rule:

   REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe

   Hits:

   0

Content Switching

In today’s complex Web sites, you may want to present different content to different users. For example, you may want to allow users from the IP range of a customer or partner to have access to a special Web portal. You may want to present content relevant to a specific geographical area to users from that area. You may want to present content in different languages to the speakers of those languages. You may want to present content tailored to specific devices, such as smartphones, to those who use the devices. The Citrix® NetScaler® content switching feature enables the NetScaler appliance to distribute client requests across multiple servers on the basis of specific content that you wish to present to those users.

To configure content switching, first create a basic content switching setup, and then customize it to meet your needs. This entails enabling the content switching feature, setting up load balancing for the server or servers that host each version of the content that is being switched, creating a content switching virtual server, creating policies to choose which requests are directed to which load balancing virtual server, and binding the policies to the content switching virtual server. You can then customize the setup to meet your needs by setting precedence for your policies, protecting your setup by configuring a backup virtual server, and improving the performance of your setup by redirecting requests to a cache.

How Content Switching Works

Content Switching enables the NetScaler appliance to direct requests sent to the same Web host to different servers with different content. For example, you can configure the appliance to direct requests for dynamic content (such as URLs with a suffix of .asp, .dll, or .exe) to one server and requests for static content to another server. You can configure the appliance to perform content switching based on TCP/IP headers and payload.

You can also use content switching to configure the appliance to redirect requests to different servers with different content on the basis of various client attributes. Some of those client attributes are:

- **Device Type.** The appliance examines the user agent or custom HTTP header in the client request for the type of device from which the request originated. Based on the device type, it directs the request to a specific Web server. For example, if the request came from a cell phone, the request is directed to a server that is capable of serving content that the user can view on his or her cell phone. A request from a computer is directed to a different server that is capable of serving content designed for a computer screen.

- **Language.** The appliance examines the Accept-Language HTTP header in the client request and determines the language used by the client's browser. The appliance then sends the request to a server that serves content in that language. For example, using content switching based on language, the appliance can send someone whose browser is configured to request content in French to a server with the French version of a newspaper. It can send someone else whose browser is configured to request content in English to a server with the English version.

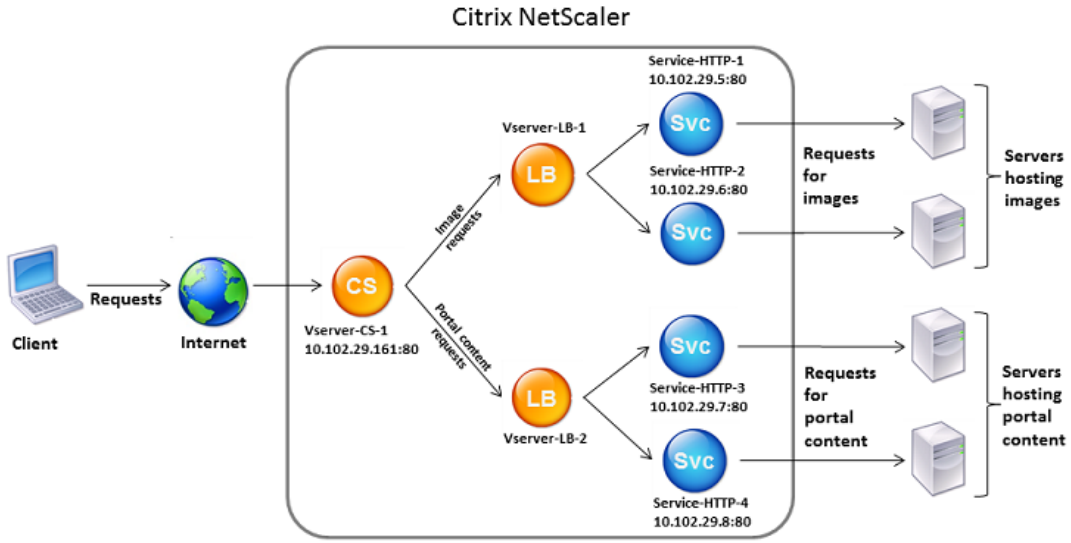
- **Cookie.** The appliance examines the HTTP request headers for a cookie that the server set previously. If it finds the cookie, it directs requests to the appropriate server, which hosts custom content. For example, if a cookie is found that indicates that the client is a member of a customer loyalty program, the request is directed to a faster server or one with special content. If it does not find a cookie, or if the cookie indicates that the user is not a member, the request is directed to a server for the general public.

- **HTTP Method.** The appliance examines the HTTP header for the method used, and sends the client request to the right server. For example, GET requests for images can be directed to an image server, while POST requests can be directed to a faster server that handles dynamic content.

- **Layer 3/4 Data.** The appliance examines requests for the source or destination IP, source or destination port, or any other information present in the TCP or UDP headers, and directs the client request to the right server. For example, requests from source IPs that belong to customers can be directed to a custom web portal on a faster server, or one with special content.

A typical content switching deployment consists of the entities described in the following diagram.

Figure 1. Content Switching Architecture



A content switching configuration consists of a content switching virtual server, a load balancing setup consisting of load balancing virtual servers and services, and content switching policies. To configure content switching, you must configure a content switching virtual server and associate it with policies and load balancing virtual servers. This process creates a *content group*—a group of all virtual servers and policies involved in a particular content switching configuration.

Content switching can be used with HTTP, HTTPS, TCP, and UDP connections. For HTTPS, you must enable SSL Offload.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. The priority of the policy defines the order in which the policies bound to the content switching virtual server are evaluated. If you are using default syntax policies, when you bind a policy to the content switching virtual server, you must assign a priority to that policy. If you are using NetScaler classic policies, you can assign a priority to your policies, but are not required to do so. If you assign priorities, the policies are evaluated in the order that you set. If you do not, the NetScaler appliance evaluates your policies in the order in which they were created.

Note: In addition to configuring policy priorities, you can manipulate the order of policy evaluation by using Goto expressions and policy bank invocations. For more details about default syntax policy configuration, see "Configuring Default Syntax Policies" in *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

After it evaluates the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

Content switching virtual servers can only send requests to other virtual servers. If you are using an external load balancer, you must create a load balancing virtual server for it and bind its virtual server as a service to the content switching virtual server.

How Content Switching Works

Content Switching enables the NetScaler appliance to direct requests sent to the same Web host to different servers with different content. For example, you can configure the appliance to direct requests for dynamic content (such as URLs with a suffix of .asp, .dll, or .exe) to one server and requests for static content to another server. You can configure the appliance to perform content switching based on TCP/IP headers and payload.

You can also use content switching to configure the appliance to redirect requests to different servers with different content on the basis of various client attributes. Some of those client attributes are:

- **Device Type.** The appliance examines the user agent or custom HTTP header in the client request for the type of device from which the request originated. Based on the device type, it directs the request to a specific Web server. For example, if the request came from a cell phone, the request is directed to a server that is capable of serving content that the user can view on his or her cell phone. A request from a computer is directed to a different server that is capable of serving content designed for a computer screen.

- **Language.** The appliance examines the Accept-Language HTTP header in the client request and determines the language used by the client's browser. The appliance then sends the request to a server that serves content in that language. For example, using content switching based on language, the appliance can send someone whose browser is configured to request content in French to a server with the French version of a newspaper. It can send someone else whose browser is configured to request content in English to a server with the English version.

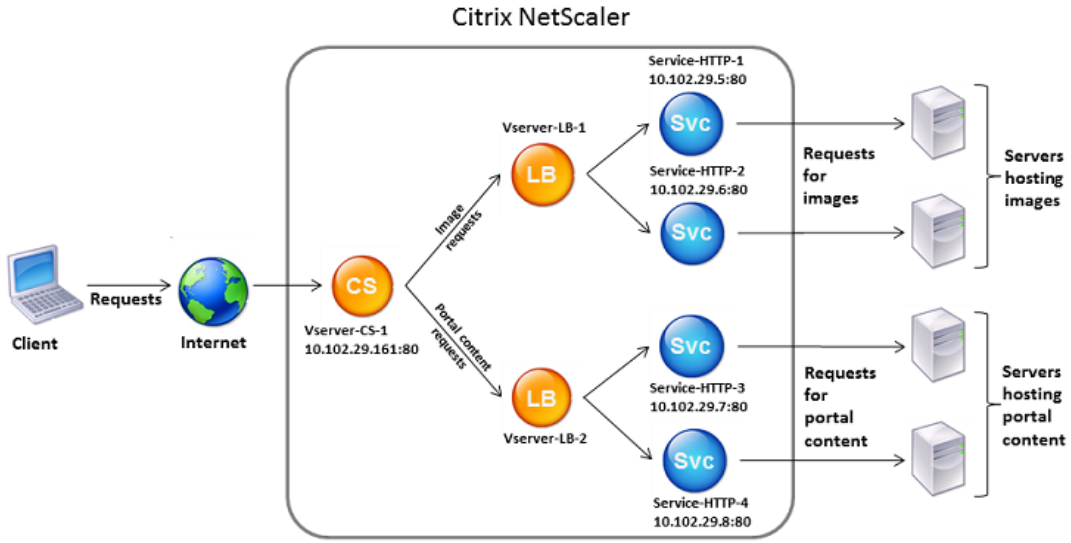
- **Cookie.** The appliance examines the HTTP request headers for a cookie that the server set previously. If it finds the cookie, it directs requests to the appropriate server, which hosts custom content. For example, if a cookie is found that indicates that the client is a member of a customer loyalty program, the request is directed to a faster server or one with special content. If it does not find a cookie, or if the cookie indicates that the user is not a member, the request is directed to a server for the general public.

- **HTTP Method.** The appliance examines the HTTP header for the method used, and sends the client request to the right server. For example, GET requests for images can be directed to an image server, while POST requests can be directed to a faster server that handles dynamic content.

- **Layer 3/4 Data.** The appliance examines requests for the source or destination IP, source or destination port, or any other information present in the TCP or UDP headers, and directs the client request to the right server. For example, requests from source IPs that belong to customers can be directed to a custom web portal on a faster server, or one with special content.

A typical content switching deployment consists of the entities described in the following diagram.

Figure 1. Content Switching Architecture



A content switching configuration consists of a content switching virtual server, a load balancing setup consisting of load balancing virtual servers and services, and content switching policies. To configure content switching, you must configure a content switching virtual server and associate it with policies and load balancing virtual servers. This process creates a *content group*—a group of all virtual servers and policies involved in a particular content switching configuration.

Content switching can be used with HTTP, HTTPS, TCP, and UDP connections. For HTTPS, you must enable SSL Offload.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. The priority of the policy defines the order in which the policies bound to the content switching virtual server are evaluated. If you are using default syntax policies, when you bind a policy to the content switching virtual server, you must assign a priority to that policy. If you are using NetScaler classic policies, you can assign a priority to your policies, but are not required to do so. If you assign priorities, the policies are evaluated in the order that you set. If you do not, the NetScaler appliance evaluates your policies in the order in which they were created.

Note: In addition to configuring policy priorities, you can manipulate the order of policy evaluation by using Goto expressions and policy bank invocations. For more details about default syntax policy configuration, see "Configuring Default Syntax Policies" in *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

After it evaluates the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

Content switching virtual servers can only send requests to other virtual servers. If you are using an external load balancer, you must create a load balancing virtual server for it and bind its virtual server as a service to the content switching virtual server.

Configuring Basic Content Switching

Before you configure content switching, you must understand how content switching is set up and how the services and virtual servers are connected.

To configure a basic, functional content switching setup, first enable the content switching feature. After you enable content switching, create a content switching virtual server to accept requests to those of your Web sites that use content switching. After you create a content switching virtual server, create a load balancing setup. Then, create two or more policies that select requests for content switching, and bind those policies to the content switching virtual server. When you bind a policy, you specify the load balancing virtual server to which requests that match the policy are to be directed.

Note: This information covers creation of a new content switching configuration. For information on modifying the configuration of an existing content switching virtual server, see Customizing the Basic Content Switching Configuration. For information on disabling and re-enabling entities, unbinding policies, and removing entities, see Managing a Content Switching Setup.

Understanding the Topology

In a content switching setup, the NetScaler appliances are logically located between the client and the server farm. The following diagram shows the topology of a basic content switching configuration.

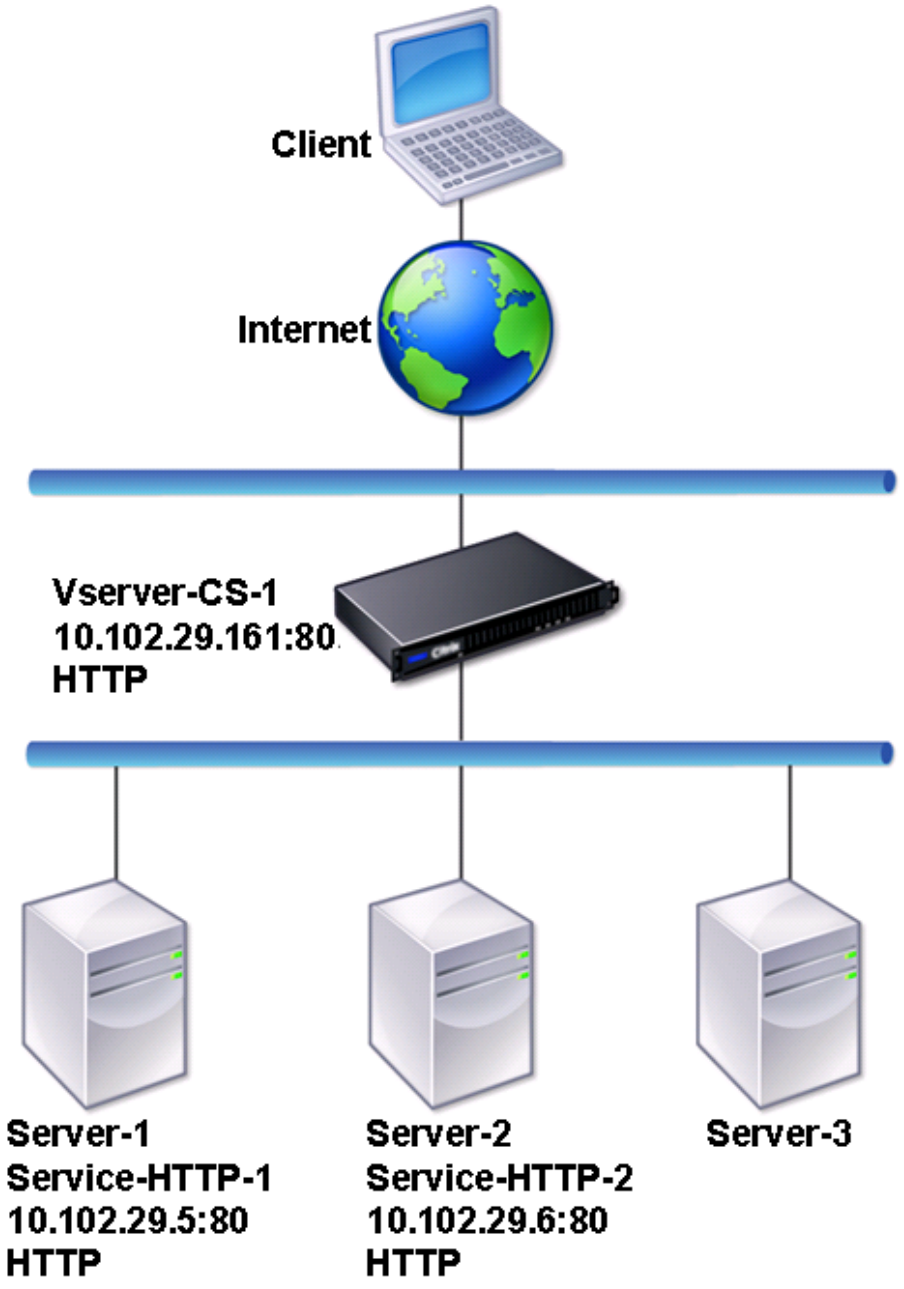


Figure 1. Topology of a Basic Content Switching Configuration

Policies manage the traffic flow. Requests that match a specific policy are directed to a specific load balancing virtual server. You therefore need to know how to configure a load balancing setup. For information about load balancing, see Load Balancing. In the following example, content switching virtual server Vserver-CS-1 is configured to send certain types of requests to load balancing virtual server Vserver-LB-1. Vserver-LB-1 balances the load across the services bound to Vserver-LB-1. The following table lists the names and values of the basic entities that must be configured on the appliance.

Table 1. Sample Content Switching Configuration

Entity Type	Mandatory Parameters and Sample Values			
	Name	**IP address**	**Port**	**Protocol**
Virtual servers	Vserver-CS-1	10.102.29.161	80	HTTP
	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	8083	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP
Monitors	Default	None	None	None

The following diagram shows the content switching sample values and mandatory parameters that are described in the preceding table.

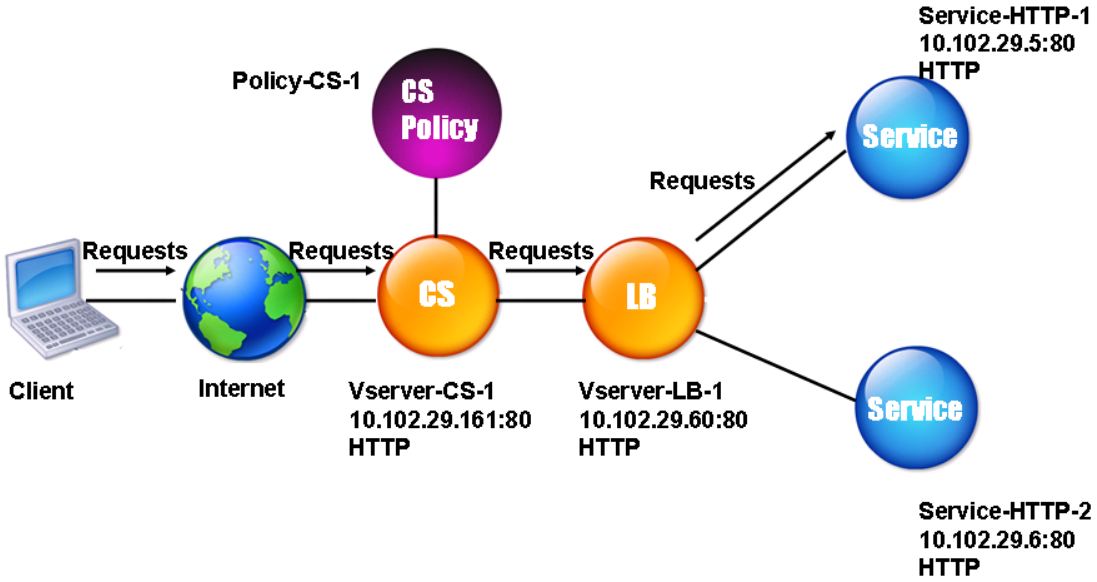


Figure 2. Content Switching Entity Model

Enabling Content Switching

To use the content switching feature, you must enable content switching. You can configure content switching entities even though the content switching feature is disabled. However, the entities will not work.

To enable content switching by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable content switching and verify the configuration:

- enable feature CS
- show feature

Example

```
> enable  feature ContentSwitch
 Done
> show feature

      Feature                 Acronym           Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               ON
3)    Load Balancing           LB               ON
4)    Content Switching          CS               ON
.
.
.
22)   Responder              RESPONDER          ON
23)   HTML Injection          HTMLInjection       ON
24)   NetScaler Push          push             OFF
Done
```

To enable content switching by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Content Switching** check box, and then click **OK**.
4. In the **Enable/Disable Features(?)** dialog box, click **Yes**.

Creating Content Switching Virtual Servers

You can add, modify, and remove content switching virtual servers. The state of a virtual server is DOWN when you create it, because the load balancing virtual server is not yet bound to it.

To create a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

add cs vserver <vServerName> <protocol> <IPAddress> <port>

Example

add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80

Parameters for configuring content switching

vServerName

Name of the content switching virtual server. This alphanumeric string is required and cannot be changed after the content switching virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and blank space.

IPAddress

IP address of the virtual server. This IP address (VIP) is usually a public IP address to which the clients send connection requests.

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

protocol

Protocol of the requests processed by the content switching virtual server. Choose one of the following service types:

- **HTTP.** For HTTP services.
- **TCP.** For non-RFC implementation of HTTP services.
- **UDP.** For DNS, ICMP, and other UDP-based services.
- **FTP.** For FTP services. This setting ensures that the NetScaler appliance takes care of the specifics of the FTP protocol.
- **SSL.** For HTTPS services. Select this type to encrypt HTTP traffic between the NetScaler appliance and the server.
- **SSL_TCP.** For secure TCP services.
- **RTSP.** For Real-Time Streaming Protocol services.
- **DNS.** For name servers.
- **SIP-UDP.** For SIP servers.
- **RDP.**
- **ANY.** For services that accept all traffic of any protocol type.

To add a content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Content Switching)** dialog box, in the **Name**, **IP Address**, and **Port** text boxes, type the name, IP address, and port of the virtual server, (for example, **Vserver-CS-1**, **10.102.29.161**, and **80**).

   Note: If you need to enter an IPv6 address, select the **IPv6** check box before you enter the address.
4. In the **Protocol** list, select the type of the virtual server (for example, **HTTP**).
5. Click **Create**, and then click **Close**.

Configuring a Load Balancing Setup for Content Switching

The content switching virtual server redirects all requests to a load balancing virtual server. You must create one load balancing virtual server for each version of the content that is being switched. This is true even when your setup has only one server for each version of the content, and you are therefore not doing any load balancing with those servers. You can also configure actual load balancing with multiple load-balanced servers that mirror each version of the content. In either scenario, the content switching virtual server needs to have a specific load balancing virtual server assigned to each version of the content that is being switched.

The load balancing virtual server then forwards the request to a service. If it has only one service bound to it, it selects that service. If it has multiple services bound to it, it uses its configured load balancing method to select a service for the request, and forwards that request to the service that it selected.

To configure a basic load balancing setup, you need to perform the following tasks:

- Create load balancing virtual servers
- Create services
- Bind services to the load balancing virtual server

For more information on load balancing, see Load Balancing. For detailed instructions on setting up a basic load balancing configuration, see Setting Up Basic Load Balancing.

Creating Content Switching Policies

A content switching policy defines a type of request that is to be directed to a load balancing virtual server. These policies are applied in the order of the priorities assigned to them or (if you are using NetScaler classic policies and do not assign priorities when binding them) in the order in which the policies were created.

The policies can be:

- **Domain-based policies.** The NetScaler appliance compares the domain of an incoming URL with the domains specified in the policies. The appliance then returns the most appropriate content. Domain-based policies must be classic policies; default syntax policies are not supported for this type of content switching policy.
- **URL-based policies.** The appliance compares an incoming URL with the URLs specified in the policies. The appliance then returns the most appropriate URL-based content, which is usually the longest matching configured URL. URL-based policies must be classic policies; default syntax policies are not supported for this type of content switching policy.

  Rule-based policies. The appliance compares incoming data to expressions specified in the policies. You create rule-based policies by using either a classic expression or a default syntax expression. Both classic and default syntax policies are supported for rule-based content switching policies.

  If you set a priority when binding your policies to the content switching virtual server, the policies are evaluated in order of priority. If you do not set specific priorities when binding your policies, the policies are evaluated in the order in which they were created.

For information about NetScaler classic policies and expressions, see the "Configuring Classic Policies and Expressions" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673. For information about Default Syntax policies, see the "Configuring Default Syntax Expressions" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To create a content switching policy by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- add cs policy <policyName> -**domain** <domain>
- add cs policy <policyName> -**url** <URLValue>

- add cs policy <policyName> -**rule** <RULEValue>

Example

add cs policy Policy-CS-1 -url "/sports/*"

add cs policy Policy-CS-1 -domain "example.com"

add cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(24).EQ(10.217.84.0)"

add cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009 Dec)"

Parameters for configuring content switching policies

policyName

Name of the content switching policy. This alphanumeric string is required and cannot be changed after the content switching policy is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

domain

The domain in the URL. The NetScaler appliance uses the domain to choose the correct content for each incoming request.

URLValue

An absolute or relative URL. The NetScaler appliance uses the URL to choose the correct content for each incoming request.

RULEValue

A classic or Default Syntax policy expression that defines the appropriate content for each request.

To create a content switching policy by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create Content Switching Policy** dialog box, in the **Name** text box, type the name of the policy (for example, **Policy-CS-1**).
4. Choose the type of policy that you want to create, and configure the policy.
   - To create a domain-based policy, in the **Domain** text box, type the domain (for example, **example.com**).
   - To create a URL-based policy, click **URL**, and in the **Value** text box, type an absolute or relative URL (for example, **http://www.example.com/sports**, or just **/sports**).
   - To create a rule-based policy, click **Configure**, and do the following:

     In the **Create Expression** dialog box, choose the expression syntax you want to use.

     - If you want to use default syntax, accept the default and proceed to the next step.

       If you want to use classic syntax, click **Switch to Classic Syntax**.

       The **Expression** portion of the dialog box changes to match your choice. The default syntax **Expression** view has fewer elements than does the classic syntax **Expression** view. In the default syntax **Expression** view, instead of a preview window, a button provides access to an expression evaluator. The evaluator evaluates the expression you entered, to verify that it is valid, and displays an analysis of the expression's effect.

     Enter your policy expressions.

     - If you are using classic syntax and need further instructions, see the "Configuring Classic Policies and Expressions" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.
     - If you are using the default syntax and need further instructions, see the "Configuring Default Syntax Expressions" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.
5. Click **Create**, and then click **Close**. The policy you created appears in the **Content Switching Policies** pane.

Configuring Content Switching Policy Labels

A policy label is a user-defined bind point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority you configured. A policy label can include one or more policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and return of control to the policy that invoked the policy label. You can create policy labels only for advanced policies.

For information about policy labels, see the "Creating Policy Labels" chapter in the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

A content switching policy label consists of a name, a label type (which is the protocol that is configured for the policies bound to the policy label), and a list of policies bound to the policy label. The policy label type must match the service type (protocol) of the content switching virtual server to which the policy that invokes the policy label is bound. For example, you can bind TCP Payload policies to a policy label of type TCP only. Binding TCP Payload policies to a policy label of type HTTP is not supported.

Each policy in a content switching policy label is associated with either a target (which is equivalent to the action that is associated with other types of policies, such as rewrite and responder policies) or a gotoPriorityExpression option and/or an invoke option. That is, for a given policy in a content switching policy label, you can specify a target, or you can set the gotoPriorityExpression option and/or the invoke option. Additionally, if multiple policies evaluate to true, only the target of the last policy that evaluates to true is considered.

You can use either the NetScaler command line or the configuration utility to configure content switching policy labels. In the NetScaler command-line interface (CLI), you first create a policy label by using the add cs policylabel command. Then, you bind policies to the policy label, one policy at a time, by using the bind cs policylabel command. In the NetScaler configuration utility, you perform both tasks in a single dialog box.

To create a content switching policy label by using the NetScaler command line

At the NetScaler command prompt, type:

add cs policylabel <labelNamename> <cspolicylabelTypetype>

Example

add cs policylabel testpollab http

Parameters for configuring policy labels

name

A name for your new policy label. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. This name must be unique within the list of policy labels for Content Switching.

type

The protocol supported by the policy label. Your choices are:

- **HTTP**. The default choice. Supports policies that process HTTP traffic. This is the protocol used to access unencrypted Web sites.
- **SSL**. Supports policies that process HTTPS/SSL encrypted traffic. This is the protocol used to access encrypted Web sites.
- **TCP**. Supports policies that process any type of TCP traffic, including HTTP.
- **SSL_TCP**. Supports policies that process SSL-encrypted TCP traffic, including SSL.
- **UDP**. Supports policies that process any type of UDP-based traffic, including DNS.
- **DNS**. Supports policies that process DNS traffic.
- **ANY**. Supports all types of policies.
- **SIP_UDP**. Supports policies that process UDP-based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP).
- **RTSP**. Supports policies that process Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data, such as audio, video, and other types of streamed media.
- **RADIUS**. Supports policies that process Remote Authentication Dial In User Service (RADIUS) traffic. RADIUS supports combined authentication, authorization, and auditing services for network management.

All policies bound to the policy label must either match the designated protocol or be a subtype of the designated protocol.

To bind a policy to a policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a policy to a policy label and verify the configuration:

- bind cs policylabel <labelName> <policyName> <priority> [**-targetVserver** <string>] [**-gotoPriorityExpression** <expression>][**-invoke** <labeltype> <labelName>]
- show cs policylabel <labelname>

Example

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 1
        Number of times invoked: 0
1)      Policy Name: test_Pol
        Priority: 100
        Target Virtual Server: LBVIP
```

To unbind a policy from a policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a policy from a policy label and verify the configuration:

- unbind cs policylabel <labelname> <policyName>
- show cs policylabel <labelname>

Example

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
```

To remove a policy label by using the NetScaler command line

At the NetScaler command prompt, type:

rm cs policylabel <name>

Parameters for configuring a content switching policy label

labelName:

The name of the content switching policy label. Maximum length: 127 characters.

cspolicylabeltype:

The type of the policy label. Possible values: HTTP, TCP, RTSP, SSL, SSL_TCP, UDP, DNS, SIP_UDP, ANY, RADIUS.

policyName:

The name of the policy to be bound to the content switching policy label. Maximum Length: 127 characters.

priority:

The priority with which the policy is to be bound. Minimum value: 1; Maximum value: 2147483647.

targetVserver:

The name of the load balancing virtual server to which requests that match the policy are switched. Maximum Length: 127 characters.

gotoPriorityExpression:

The priority number of the next expression to be evaluated. Goto can only proceed forward in a policy label. As an alternative to specifying the priority number of the next expression to be evaluated, you can accept the default or specify one of the following values:

- **NEXT**. Go to the policy with the next higher priority.
- **END**. Stop evaluation. (This is the default)
- **USE_INVOCATION_RESULT**. Applicable if this entry invokes another policy label. If the final Goto in the invoked policy label has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy label performs a NEXT.

invoke:

You can invoke other policy labels or virtual servers by using this option. After evaluating the policies in the invoked policy label, the appliance continues to evaluate policies that are bound to the current policy label (the selected bind point).

To configure a content switching policy label by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policy Labels**.
2. In the details pane, do one of the following:
   - To create a new policy label, click **Add**.
   - To modify an existing policy label, select the policy label, and then click **Open**.
3. If you are creating a new policy label, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring policy labels" as shown:
   - **Name***—name (Cannot be changed for an existing policy)
   - **Label Type***—type (Cannot be changed for an existing policy)

   * A required parameter
4. To add a policy to a list, click **Insert Policy**, and then click one of the policies in the drop-down list. If you click **New Policy**, create a new policy as described in Creating Content Switching Policies. After you add a policy to the list, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a content switching policy label" as shown:
   - **Priority***—priority (The default value is 100. To modify the value, double-click in the Priority column.)
   - **Target**—targetVserver
   - **Goto Expression**—gotoPriorityExpression
   - **Invoke**—invoke

   * A required parameter
5. To remove a policy from the list, select that policy, and then click **Unbind Policy**.
6. To automatically renumber the policies, click **Regenerate Priorities**.
7. To modify a policy, click the name of the policy, and then click **Modify Policy**.
8. To modify the target virtual server that is configured for policy, click the name of the virtual server, and then click **Modify Virtual Server**.
9. To modify the policy label that a policy is configured to invoke, click the name of the policy label, and then click **Modify Invoke Label**.
10. Click **Create** or **OK**.

Binding Policies to a Content Switching Virtual Server

After you create your content switching virtual server and policies, you bind each policy to the content switching virtual server. You also select a load balancing virtual server as the target for the policy so that, after the content switching virtual server evaluates the policy, it routes requests that matches the policy to the load balancing virtual server for forwarding to the appropriate content server.

To bind a policy to a content switching virtual server and select a load balancing virtual server target by using the NetScaler command line

At the NetScaler command prompt, type:

bind cs vserver <ContentSwitchingVirtualServerName> <TargetLoadBalancingVirtualServerName> -**policyname** <PolicyName> -**priority** <PositiveInteger>

Example

bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -priority 20

To bind a policy to a content switching virtual server and select a load balancing virtual server target by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, double-click the virtual server for which you want to bind the policy (for example, **Vserver-CS-1**).
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Policies** tab, in the **Active** column, select the **Active** check box next to the policy that you want to bind to the virtual server (for example, **Policy-CS-1**).
4. In the **Target** column next to the policy, select the load balancing virtual server that you want to assign as the target for the policy (for example, **Vserver-LB-1**).
5. Click **OK**.

Verifying the Configuration

To verify that your content switching configuration is correct, you need to view the content switching entities. To verify proper operation after your content switching configuration has been deployed, you can view the statistics that are generated as the servers are accessed.

Viewing the Properties of Content Switching Virtual Servers

You can view the properties of content switching virtual servers that you have configured on the NetScaler. You can use the information to verify whether the virtual server is correctly configured and, if necessary, to troubleshoot. In addition to details such as name, IP address, and port, you can view the various policies bound to a virtual server, and its traffic-management settings.

The content switching policies are displayed in the order of their priority. If more than one policy has the same priority, they are shown in the order in which they are bound to the virtual server.

Note: If you have configured the content switching virtual server to forward traffic to a load balancing virtual server, you can also view the content switching policies by viewing the properties of the load balancing virtual server.

To view the properties of content switching virtual servers by using the NetScaler command line

To list basic properties of all content switching virtual servers in your configuration, or detailed properties of a specific content switching virtual server, at the NetScaler command prompt, type one of the following commands:

- show cs vserver
- show cs vserver <vServerName>

Example

1.
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE

Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

1) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

1) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

1) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
GotoPriority Expression: END

1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
Done
>

2.
show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
…

To view the properties of content switching virtual servers by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, click a virtual server to display its configuration details at the bottom of the screen.
3. To display the names of the policies that are bound to the content switching virtual server, double-click the virtual server and then click the **Policies** tab.

Viewing Content Switching Policies

You can view the properties of the content switching policies that you defined, such as the name, domain, and URL or expression, and use the information to find any mistakes in the configuration, or to troubleshoot if something is not working as it should.

To view the properties of content switching policies by using the NetScaler command line

To list either basic properties of all content switching policies in your configuration or detailed properties of a specific content switching policy, at the NetScaler command prompt, type one of the following commands:

- show cs policy
- show cs policy <PolicyName>

Example

show cs policy

show cs policy Policy-CS-1

To view the properties of content switching policies by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, double-click a policy to view the details.
3. To view the policy labels and virtual servers that this policy is bound to, on the **Content Switching Policies** pane, click **Show Bindings**.

Viewing a Content Switching Virtual Server Configuration by Using the Visualizer

The Content Switching Visualizer is a tool that you can use to view a content switching configuration in graphical format. You can use the visualizer to view the following configuration items:

- A summary of the load balancing virtual servers to which the content switching virtual server is bound.
- All services and service groups that are bound to the load balancing virtual server and all monitors that are bound to the services.
- The configuration details of any displayed element.
- Any policies bound to the content switching virtual server. These policies need not be content switching policies. Many types of policies, such as Rewrite policies, can be bound to a content switching virtual server.

After you configure the various elements in a content switching and load balancing setup, you can export the entire configuration to an application template file.

Note: The Visualizer requires a graphical interface, so it is available only through the configuration utility.

To view a content switching configuration by using the Visualizer in the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the **Content Switching Visualizer** window, you can adjust the viewable area as follows:
   - Click the **Zoom In** and **Zoom Out** icons to increase or decrease the viewable area.
   - Click the **Save Image** icon to save the graph as an image file.
   - In the **Search in** text field, begin typing the name of the item you are looking for. When you have typed enough characters to identify the item, its location is highlighted. To restrict the search, click the drop-down menu and select the type of element that you want to search for.
4. To view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view policies that are bound to the virtual server, in the tool bar at the top of the dialog box select one or more feature-specific policy icons. If policy labels are configured, they appear in the main view area.
   - To view the configuration details for a bound service or service group, click the icon for the service, click the **Related Tasks** tab, and then click **Show Member Services**.
   - To view the configuration details for a monitor, click the icon for the monitor, click the **Related Tasks** tab, and then click **View Monitor**.
5. To view detailed statistics for any virtual server in the content switching configuration, click the virtual server for which you want to view statistics, then click the **Related Tasks** tab, and then click **Statistics**.
6. To view a comparative list of the parameters whose values either differ or are not defined across service containers for a load balancing virtual server, click the icon for a container, click the **Related Tasks** tab, and then click **Service Attributes Diff**.
7. To view monitor binding details for the services in a container, in the **Service Attributes Diff** dialog box, in the **Group** column for the container, click **Details**. This comparative list helps you determine which service container has the configuration you want to apply to all the service containers.
8. To view the number of requests received per second at a given point in time by the virtual servers in the configuration, and the number of hits per second at a given point in time for rewrite, responder, and cache policies, click **Show Stats**. The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time. It has to be refreshed manually. To refresh the information, click **Refresh Stats**.

   Note: The Show Stats option is available from the Visualizer only on the NetScaler 9.2 nCore and 9.3 nCore.

9. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click **Related Tasks**, click **Copy Properties**, and then paste the information into a document.

10. To export the entire configuration that is displayed in the Visualizer to an application template file, click the icon for the content switching virtual server, click **Related Tasks**, and then click **Create Template**. When creating the application template, you can configure variables in some policy expressions and actions. For more information about creating the application template file and configuring variables for a template, see the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Customizing the Basic Content Switching Configuration

After you configure a basic content switching setup, you may need to customize it to meet your requirements. If your Web servers are UNIX-based and rely on case sensitive pathnames, you can configure case sensitivity for policy evaluation. You can also set precedence for evaluation of the content switching policies that you configured. If you want to configure content switching for a specific a virtual LAN, you can configure a content switching virtual server with a listen policy.

Configuring Case Sensitivity for Policy Evaluation

You can configure the content switching virtual server to treat URLs as case sensitive in URL-based policies. When case sensitivity is configured, the NetScaler appliance considers case when evaluating policies. For example, if case sensitivity is off, the URLs /a/1.htm and /A/1.HTM are treated as identical. If case sensitivity is on, those URLs are treated as separate and can be switched to different targets.

To configure case sensitivity by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**caseSensitive** (**ON**|**OFF**)

Example

set cs vserver Vserver-CS-1 -caseSensitive ON

Parameters for configuring case sensitivity

vServerName

The name of the content switching virtual server that you are configuring. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

caseSensitive

The URL lookup case option on the content switching vserver. If case sensitivity of a content switching virtual server is set to 'ON', the URLs /a/1.html and /A/1.HTML are treated differently and may have different targets (set by content switching policies).

If case sensitivity is set to 'OFF', the URLs /a/1.html and /A/1.HTML are treated the same, and will be switched to the same target.

Possible values: ON, OFF

Default value: ON

To configure case sensitivity by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure case sensitivity (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Advanced** tab, select **Case Sensitivity** check box, and then click **OK**.

Setting the Precedence for Policy Evaluation

Precedence refers to the order in which policies that are bound to a virtual server are evaluated. You do not normally have to configure precedence: the default precedence works correctly in many cases. If you want to make sure that one policy or set of policies is applied first, however, and another policy or set of policies is applied only if the first set does not match a request, you can configure either URL-based precedence or rule-based precedence.

Precedence with URL-Based Policies

If there are multiple matching URLs for the incoming request, the precedence (priority) for URL-based policies is:

1. Domain and exact URL
2. Domain, prefix, and suffix
3. Domain and suffix
4. Domain and prefix
5. Domain only
6. Exact URL
7. Prefix and suffix
8. Suffix only
9. Prefix only
10. Default

If you configure precedence based on URL, the request URL is compared to the configured URLs. If none of the configured URLs match the request URL, then rule-based policies are checked. If the request URL does not match any rule-based policies, or if the content group selected for the request is down, then the request is processed as follows:

- If you configure a default group for the content switching virtual server, then the request is forwarded to the default group.

- If the configured default group is down or if no default group is configured, then an “HTTP 404 Not Found” error message is sent to the client.

Note: You should configure URL-based precedence if the content type (for example, images) is the same for all clients. However, if different types of content must be served based on client attributes (such as Accept-Language), you must use rule-based precedence.

Precedence with Rule-Based Policies

If you configure precedence based on rules, which is the default setting, the request is tested on the basis of the rule-based policies you have configured. If the request does not match any rule-based policies, or if the content group selected for the incoming request is down, the request is processed in the following manner:

- If a default group is configured for the content switching virtual server, the request is forwarded to the default group.
- If the configured default group is down or if no default group is configured, an “HTTP 404 Not Found” error message is sent to the client.

To configure precedence by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**precedence** (**RULE** | **URL**)

Example

set cs vserver Vserver-CS-1 -precedence RULE

Parameters for configuring precedence

vServerName

The name of the content switching virtual server that you are configuring. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

precedence

The type of precedence to use for both RULE-based and URL-based policies on the content switching virtual server. With the precedence set to RULE, incoming requests are evaluated against the rule-based content switching policies. If none of the rules match, the URL in the request is evaluated against the URL-based content switching policies. Possible values: RULE, URL. Default: RULE.

To configure precedence by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure precedence, (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Advanced** tab, under **Precedence**, click **Rule** or **URL**, and then click **OK**.

Configuring per-VLAN Wildcarded Virtual Servers

If you want to configure content switching for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the NetScaler command line

At the NetScaler command prompt, type:

add cs vserver <name> serviceType <type> IPAddress * Port * -**listenpolicy** <expression> [-**listenpriority** <positive_integer>]

Example

```
add cs vserver Vserver-CS-vlan1 ANY * *
    -listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
```

Parameters for configuring per-VLAN wildcarded virtual servers

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

IPAddress

IP address of the virtual server. For wildcarded virtual servers bound to VLANs, this is always *.

type

Behavior of the service. Select one of the following service types: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP.

port

Port on which the virtual server listens for client connections. The port number must be in the range 0-65535. For wildcarded virtual servers bound to VLANs, the setting is normally *.

listenpriority

The priority assigned to the listen policy. This can be any positive integer. Priority is evaluated in reverse order; the lower the number, the higher the priority assigned to the listen policy.

rule

The policy rule to use to identify the VLAN that you want this virtual server to listen to. This rule is:

CLIENT.VLAN.ID.EQ(<integer>)

For <integer>, substitute the ID number assigned to the VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, do one of the following:
   - To create a new virtual server, click **Add**.
   - To modify an existing virtual server, select the virtual server, and then click **Open**.
3. In the **Create Virtual Server** or **Configure Virtual Server** dialog box, on the **Services** tab, type or select values for the following parameters, which correspond to parameters described in "Parameters for configuring per-VLAN wildcarded virtual servers" as shown:
   - **Name***—name (Cannot be changed for a previously configured virtual server)
   - **Protocol***—type
   - **IP address***—IPAddress
   - **Port**—port

   * A required parameter
4. In the **Advanced** tab, expand **Listen Policy**, and then type or select values for the following parameters, which correspond to parameters described in "Parameters for configuring per-VLAN wildcarded virtual servers" as shown:
   - **Listen Priority***—priority
   - **Listen Policy Rule***—rule

   * A required parameter
5. Click **Create** or **OK**, depending on whether you are creating a new virtual server or modifying an existing virtual server.
6. Click **Close**. The virtual server that you created now appears in the Virtual Servers page.
7. To remove a virtual server, in the **Virtual Servers** pane select the virtual server, and then click **Remove**.

After you have created this virtual server, you bind it to one or more services as described in Binding Services to the Virtual Server.

Protecting the Content Switching Setup against Failure

Content switching may fail when the content switching virtual server goes DOWN or fails to handle excessive traffic, or for other reasons. To reduce the chances of failure, you can take the following measures to protect the content switching setup against failure:

- Configure a backup content switching virtual server.
- Configure spillover for preventing the overloading of the primary and diverting excess traffic to the backup virtual server.
- Specify a redirect URL, the URL to which the content is switched if both the primary and backup content switching virtual servers are DOWN.
- Enable the State Update option for marking a content switching virtual server as DOWN when the load balancing virtual server is DOWN.
- Flush the surge queues when the queues become too long.

Configuring a Redirection URL

You can configure a redirect URL to communicate the status of the NetScaler appliance in the event that a content switching virtual server of type HTTP or HTTPS is DOWN or DISABLED. This URL can be local or remote.

Redirect URLs can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Note: If a content switching virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect URL is used when the primary and backup virtual servers are down.

When redirection is configured and the content switching virtual server is unavailable, the appliance issues an HTTP 302 redirect to the user's browser.

To configure a redirect URL for when the content switching virtual server is unavailable by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**redirectURL** <URLValue>

Example

set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/mysite/maintenance

Parameters for configuring a redirect URL

vServerName

The name of the content switching virtual server that you are configuring. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

URLValue

URL to which traffic is redirected if the content switching virtual server becomes unavailable. This value must not exceed 127 characters. The domain specified in the URL must not match the domain specified in the domain name argument of a content switching policy. If the same domain is specified in both arguments, the request is redirected continuously to the same unavailable virtual server in the NetScaler, and the user cannot get the requested content.

To configure a redirect URL for when the content switching virtual server is unavailable by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure a redirect URL (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Advanced** tab, in the **Redirect URL** text box, type the redirect URL (for example, **http://www.newdomain.com/mysite/maintenance**).
4. Click **OK**.

Configuring a Backup Virtual Server

If the primary content switching virtual server is marked DOWN or DISABLED, the NetScaler appliance can direct requests to a backup content switching virtual server. It can also send a notification message to the client regarding the site outage or maintenance. The backup content switching virtual server is a proxy and is transparent to the client.

When configuring the backup virtual server, you can specify the configuration parameter Disable Primary When Down to ensure that, when the primary virtual server comes back up, it remains the secondary until you manually force it to take over as the primary. This is useful if you want to ensure that any updates to the database on the server for the backup are preserved, enabling you to synchronize the databases before restoring the primary virtual server.

You can configure a backup content switching virtual server when you create a content switching virtual server or when you change the optional parameters of an existing content switching virtual server. You can also configure a backup content switching virtual server for an existing backup content switching virtual server, thus creating cascaded backup content switching virtual servers. The maximum depth of cascaded backup content switching virtual servers is 10. The appliance searches for a backup content switching virtual server that is up and accesses that content switching virtual server to deliver the content.

Note: If a content switching virtual server is configured with both a backup content switching virtual server and a redirect URL, the backup content switching virtual server takes precedence over the redirect URL. The redirect is used when the primary and backup virtual servers are down.

To set up a backup content switching virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <primaryVServer> -**backupVserver** <backupVserver> -**disablePrimaryOnDown** (**ON**|**OFF**)

Example

set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -disablePrimaryOnDown ON

Parameters for configuring a backup virtual server

primaryVServer

The name of the primary virtual server for which you are configuring a backup. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

backupVServer

The name of the backup virtual server that you are configuring. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

You can create a virtual server and specify the name, IP address, port, and type as described in Creating Content Switching Virtual Servers. You can use the name of the content switching virtual server as a backup content switching virtual server.

disablePrimaryOnDown

Configures the appliance to leave the former primary virtual server as secondary until you manually set it to take over as the primary. Possible Values: ON, OFF. Default: OFF.

To set up a backup content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to set up a backup content switching virtual server (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, click the **Advanced** tab.
4. In the **Backup Virtual Server** list, select the backup virtual server (for example, **Vserver-CS-2**).
5. If you want to configure the backup server to remain as the primary server after the primary virtual server is brought back up, select the **Disable Primary When Down** check box.
6. Click **OK**.

Diverting Excess Traffic to a Backup Virtual Server

The spillover option diverts new connections arriving at a content switching virtual server to a backup content switching virtual server when the number of connections to the content switching virtual server exceeds the configured threshold value. The threshold value is dynamically calculated, or you can set the value. The number of established connections (in case of TCP) at the virtual server is compared with the threshold value. When the number of connections reaches the threshold, new connections are diverted to the backup content switching virtual server.

If the backup content switching virtual servers reach the configured threshold and are unable to take the load, the primary content switching virtual server diverts all requests to the redirect URL. If a redirect URL is not configured on the primary content switching virtual server, subsequent requests are dropped.

To configure a content switching virtual server to divert new connections to a backup virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**soMethod** <methodType> -**soThreshold** <thresholdValue> -**soPersistence** <persistenceValue> -**soPersistenceTimeout** <timeoutValue>

Example

set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -soPersistence enabled -soPersistenceTi

Parameters for configuring spillover

vServerName

The name of the content switching virtual server for which you are configuring spillover. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

soMethod

Type of spillover used to divert traffic to the backup content switching virtual server when the virtual server reaches the spillover threshold. The valid options for this parameter are: CONNECTION, BANDWIDTH, and NONE. For more information about how each of these methods work, see Load Balancing.

soThreshold

For the CONNECTION spillover type, the Threshold value is the maximum number of connections a virtual server can handle before spillover. For the BANDWIDTH spillover type, the Threshold value is the amount of incoming and outgoing traffic (in kilobits per second) that a virtual server can handle before spillover occurs. The minimum value is 1, and the maximum value is 4294967294.

soPersistence

The spillover persistence state. If you enable spillover persistence, the NetScaler maintains sourceIP-based persistence over primary virtual server and backup content switching virtual servers. The valid options for this parameter are: ENABLED and DISABLED. The default value is DISABLED.

soPersistenceTimeout

This value sets the timeout for spillover persistence. The default value is 2 minutes. The minimum value is 2 minutes, and the maximum value is 1440 minutes.

To set a content switching virtual server to divert new connections to a backup virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure spillover (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Advanced** tab, under **Spillover**, in the **Method** list, select the type of spillover, and in **Threshold** text box, type the threshold value (for example, **Connection** and **1000**).
4. Select the **Persistence** check box and, in **Persistence Time-out (min)** text box, type the timeout value (for example, **2**).
5. Click **OK**.

Configuring the State Update Option

The content switching feature enables the distribution of client requests across multiple servers on the basis of the specific content presented to the users. For efficient content switching, the content switching virtual server distributes the traffic to the load balancing virtual servers according to the content type, and the load balancing virtual servers distribute the traffic to the physical servers according to the specified load balancing method.

For smooth traffic management, it is important for the content switching virtual server to know the status of the load balancing virtual servers. The state update option helps to mark the content switching virtual server DOWN if the load balancing virtual server bound to it is marked DOWN. A load balancing virtual server is marked DOWN if all the physical servers bound to it are marked DOWN.

When State Update is enabled:

When you add a new content switching virtual server, initially, its status is shown as DOWN. When you bind a load balancing virtual server whose status is UP, the status of the content switching virtual server becomes UP.

When State Update is disabled:

The status of the content switching virtual server is marked as UP. It remains UP even if there is no bound load balancing virtual server that is UP.

If more than one load balancing virtual server is bound and if one of them is specified as the default, the status of the content switching virtual server reflects the status of the default load balancing virtual server.

If more than one load balancing virtual server is bound without any of them being specified as the default, the status of the content switching virtual server is marked UP only if all the bound load balancing virtual servers are UP.

To configure the state update option by using the NetScaler command line

At the NetScaler command prompt, type:

add cs vserver <vServerName> <protocol> <ipAddress> <port> **-stateUpdate** ENABLED

Example

add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED -cltTimeout 180

Parameters for configuring state update option

vServerName

Name of the content switching virtual server. This alphanumeric string is required and cannot be changed after the content switching virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and blank space.

ipAddress

IP address of the virtual server. This IP address (VIP) is usually a public IP address to which the clients send connection requests.

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

protocol

Protocol of the requests processed by the content switching virtual server. Possible values: HTTP, TCP, UDP, FTP, SSL, SSL_TCP, RTSP, DNS, SIP-UDP, ANY.

stateUpdate

Status of the content switching virtual server according to the status of the bound load balancing virtual server. Possible values: ENABLED, DISABLED. DEFAULT: DISABLED.

To configure the state update option by using the NetScaler configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, do one of the following:
   - To add a virtual server, click **Add**.
   - To modify a virtual server, select the server, and click **Open**.
3. In the **Create Virtual Server (Content Switching)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring content switching" as shown:
   - Name-vServerName
   - IP Address-ipAddress

     Note: If you need to enter an IPv6 address, select the **IPv6** check box before you enter the address.
   - Port-port
   - Protocol-protocol
4. On the **Advanced** tab, select the **State Update** check box.
5. Click **Create**.
6. Select the new virtual server, click **Open**, and verify the settings.

Managing a Content Switching Setup

After a content switching setup is configured, it may require periodic changes. When operating systems or software are updated, or hardware wears out and is replaced, you may need to take down your setup. Load on your setup may increase, requiring additional resources. You may also modify the configuration to improve performance.

These tasks may require unbinding policies from the content switching virtual server, or disabling or removing content switching virtual servers. After you have made changes to your setup, you may need to re-enable servers and rebind policies. You might also want to rename your virtual servers.

Unbinding Policies from the Content Switching Virtual Server

When you unbind a content switching policy from its virtual server, the virtual server no longer includes that policy when determining where to direct requests.

To unbind a policy from a content switching virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

unbind cs vserver <vServerName> -**policyname** <policyName>

Example

unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1

Parameters for unbinding content switching policies

vServerName

The name of the content switching virtual server from which you are unbinding the policy. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

policyName

The name of the policy that you are unbinding.

To unbind a policy from a content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server form which you want to unbind the policy (for example, **Vserver-CS-1**), and click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Policies** tab, in the **Active** column, clear the check box next to the policy that you want to unbind from the virtual server (for example, **Policy-CS-1**).
4. Click **OK**.

Removing Content Switching Virtual Servers

You normally remove a content switching virtual server only when you no longer require the virtual server. When you remove a content switching virtual server, the NetScaler appliance first unbinds all policies from the content switching virtual server, and then removes it.

To remove a content switching virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

rm cs vserver <vServerName>

Example

rm cs vserver Vserver-CS-1

To remove a content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to remove (for example, **Vserver-CS-1**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Disabling and Re-Enabling Content Switching Virtual Servers

Content switching virtual servers are enabled by default when you create them. You can disable a content switching virtual server for maintenance. If you disable the content switching virtual server, the state of the content switching virtual server changes to Out of Service. While out of service, the content switching virtual server does not respond to requests.

To disable or re-enable a virtual server by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- disable cs vserver <vServerName>
- enable cs vserver <vServerName>

Example

disable cs vserver Vserver-CS-1

enable cs vserver Vserver-CS-1

To disable or re-enable a virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to disable (for example, **Vserver-CS-1**).
3. Disable or re-enable the virtual server by clicking **Disable** or **Enable**, and then clicking **Yes** to confirm your choice.

Renaming Content Switching Virtual Servers

You can rename a content switching virtual server without unbinding it. The new name is propagated automatically to all affected parts of the NetScaler configuration.

To rename a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

rename cs vserver <vServerName> <vServerNewName>

Example

rename cs vserver Vserver-CS-1 Vserver-CS-2

To rename a virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to rename (for example, **Vserver-CS-1**).
3. Click **Rename**.
4. In the **Name** text box, type a new name for the virtual server.
5. Click **OK** to save your changes.

Managing Content Switching Policies

You can modify an existing policy by configuring rules or changing the URL of the policy, or you can remove a policy. You can create different policies based on the URL. URL-based policies can be of different types, as described in the following table.

Table 1. Examples of URL-Based Policies

Type of URL-Based Policy	Specifies
Domain and Exact URL	Requests must match the configured domain name and configured URL (an exact prefix match if only the prefix is configured; or an exact match of the prefix and suffix if both the prefix and suffix are configured). **Example:** **add cs policy Policy-CS-1 -url /sports/tennis/index.html -domain "www.domainxyz.com"**
Domain and Wild Card URL	Requests must match the exact domain name and a partial prefix of the configured URL. **Example:** **add cs policy Policy-CS-1 -url /*.jsp -domain "www.domainxyz.com"**
Domain Only	Requests need match only the configured domain name. **Example:** **add cs policy Policy-CS-1 -domain "www.domainxyz.com"**

The Exact URL	The incoming URL must exactly match the URL specified by the policy. If only a URL prefix rule is configured, there must be an exact prefix match with the incoming URL. If a URL prefix and suffix-based rule is configured, there should be an exact match of the prefix and suffix with the incoming URL. **Example:** **add cs policy Policy-CS-1 -url /sports/tennis/index.html**
Prefix Only (Wild Card URL)	All the incoming URLs must start with the configured prefix. **Example:** **add cs policy Policy-CS-1 -url /sports*** sports/*" matches all URLs under /sports "/sports*" matches all URLs whose prefix match "/sports" starting from the beginning of a URL
Suffix Only (Wild Card URL)	All incoming URLs must end with the configured URL suffix. **Example:** **add cs policy Policy-CS-1 -url /*.jsp** "/*.jsp" matches all URLs whose file extension is "jsp"
Prefix and Suffix (Wild Card URL)	All incoming URLs must start with the configured prefix and end with the configured suffix. **Example:** **add cs policy Policy-CS-1 -url /sports/*.jsp**

Note: You can configure rule-based content switching using classical policy expressions or advanced policy expressions. For more information about configuring policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To modify or remove a policy by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- set cs policy <policyName> [-**domain** <domainValue>] [-**rule** <ruleValue>] [-**url** <URLValue>]
- rm cs policy <policyName>

Example

set cs policy Policy-CS-1 -domain "www.domainxyz.com"

set cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(22).EQ(10.100.148.0)"

set cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010 Jul)"

set cs policy Policy-CS-1 -url /sports/*

rm cs policy Policy-CS-1

Parameters for configuring content switching policies

policyName

The name of the content switching policy that you are configuring. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 31 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

domain

The domain name. The alphanumeric string can range from 3 to 63 characters, and can consist of any characters that are allowed in a domain name.

rule

A NetScaler advanced policy expression that defines which requests to forward to a particular content switching virtual server.

url

A URL or partial URL that enables the NetScaler appliance to choose requests to forward to a particular content switching server.

To modify or remove a policy by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, select the policy that you want to modify (for example, **Policy-CS-1**).
3. Modify or remove the policy.
   - To modify the policy, click **Open** and then make the changes that you want. For example, you can type a new domain name in the **Domain** text box. Then, click **Yes** to confirm your changes.
   - To remove the policy, click **Remove**, and then click **Yes** to confirm your choice.
4. In the **Configure Content Switching Policies** dialog box, in the **Domain** text box, type the domain name (for example, **www.domainxyz.com**).
5. Click **OK**.

Modifying a Content Switching Configuration by Using the Visualizer

You can use the Visualizer to modify a load balancing virtual server to which the content switching virtual server is bound. You can also modify a service or group of similar services, or a monitor. For more information, see The Load Balancing Visualizer.

Managing Client Connections

To ensure efficient management of client connections, you can configure the content switching virtual servers on the NetScaler appliance to use the following features:

- Redirecting client requests to a cache
- Enabling delayed cleanup of virtual server connections
- Rewriting ports and protocols for redirection
- Inserting the IP address and port of a virtual server in the request header
- Setting a time-out value for idle client connections
- Using net profiles to specify the set of SNIPs and MIPs to be used for backend communication
- Configuring the ICMP Response. You can configure the NetScaler to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.

  The following settings can be made on a virtual server:

  - When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

Redirecting Client Requests to a Cache

The NetScaler cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the burden of responding to HTTP requests and improve your Web site performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the NetScaler appliance sends cacheable HTTP requests to the cache and non-cacheable HTTP requests to the origin Web server. For more information on cache redirection, see Cache Redirection.

To configure cache redirection on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**cacheable** <Value>

Example

set cs vserver Vserver-CS-1 -cacheable yes

Parameters for configuring cache redirection

vServerName

The name of the virtual server that you are configuring.

cacheable

Route virtual server requests to the cache redirection virtual server before sending them to the configured servers. Possible values: YES, NO. Default: NO.

To configure cache redirection on a virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure cache redirection (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Advanced** tab, select the **Cache Redirection** check box.
4. Click **OK**.

Enabling Delayed Cleanup of Virtual Server Connections

Under certain conditions, you can configure the down state flush setting to terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources and in certain cases speeds recovery of overloaded load balancing setups.

To configure the down state flush setting on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**downStateFlush** <Value>

Example

set cs vserver Vserver-CS-1 -downStateFlush enabled

Parameters for configuring down state flush

vServerName

The name of the virtual server that you are configuring.

downStateFlush

Perform delayed cleanup of connections on the virtual server. Possible values: ENABLED, DISABLED. Default: ENABLED.

To configure the down state flush setting on a virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure down state flush (for example, **Vserver-CS-1**), and click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, click the **Advanced** tab.
4. Select the **Down state flush** check box, and then click **OK**.

Rewriting Ports and Protocols for Redirection

Virtual servers and the services that are bound to them may use different ports. When a service responds to an HTTP connection with a redirect, you may need to configure the NetScaler appliance to modify the port and the protocol to ensure that the redirection goes through successfully. You do this by enabling and configuring the redirectPortRewrite setting.

To configure HTTP redirection on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**redirectPortRewrite** <Value>

Example

set cs vserver Vserver-CS-1 -redirectPortRewrite enabled

Parameters for redirect port rewrite

vServerName

The name of the virtual server that you are configuring.

redirectPortRewrite

State of port rewrite while performing HTTP redirect. Possible values: ENABLED and DISABLED. Default: DISABLED.

To configure HTTP redirection on a virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure HTTP redirection (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, click the **Advanced** tab.
4. Select the **Redirect Port Rewrite** check box, and then click **OK**.

Inserting the IP Address and Port of a Virtual Server in the Request Header

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the NetScaler appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, you must configure the required header tag on both virtual servers.

Note: This option is not supported for wildcarded virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**insertVserverIPPort** <vServerIPPORT>

Example

set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80

Parameters for virtual server IP port insertion

vServerName

The name of the virtual server that you are configuring.

insertVserverIPPort

Virtual IP address and port header insertion option for the virtual server.

VIPADDR-Header contains the virtual server IP address and port number without any translation.

If VIPADDR is not specified, the header is inserted with the name specified in the default header tag vip-header and the virtual server IP and port are inserted in the request with the default header tag vipHeader.

If VIPADDR is specified, the header is inserted with the user-specified name in vipHeader. The virtual server IP and port are inserted in the request with the user-specified header tag vipHeader.

OFF- The virtual IP and port header insertion option is disabled. The virtual server and port number are not inserted.

V6TOV4MAPPING - If the virtual server uses an IPv6 address and the server uses IPv4, this setting maps the virtual server address and port to the IPv4 address.

Possible values: OFF, VIPADDR, and V6TOV4MAPPING. Default: OFF.

To insert the IP address and port of the virtual server in the client requests by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure virtual server port insertion (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, click the **Advanced** tab.
4. In the **Vserver IP Port Insertion** list, select the **VIPADDR** or **V6TOV4MAPPING**, and then type the port header in a text box next to **Vserver IP Port Insertion** box.
5. Click **OK**.

Setting a Time-out Value for Idle Client Connections

You can configure a virtual server to terminate any idle client connections after a configured time-out period elapses. When you configure this setting, the NetScaler appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection.

To set a time-out value for idle client connections by using the NetScaler command line

At the NetScaler command prompt, type:

set cs vserver <vServerName> -**cltTimeout** <Value>

Example

set cs vserver Vserver-CS-1 -cltTimeout 100

Parameters for setting the client time-out value

vServerName

The name of the virtual server that you are configuring.

cltTimeout

Idle time (in seconds) after which the client connection is terminated. The default values are:

- 180 seconds for HTTP/SSL-based services.
- 9000 seconds for other TCP-based services.
- 180 seconds for DNS-based services.
- 180 seconds for other UDP-based services.

Maximum value: 31536000.

To set a time-out value for idle client connections by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to set a time-out value (for example, **Vserver-CS-1**), and then click **Open**.
3. In the **Configure Virtual Server (Content Switching)** dialog box, click the **Advanced** tab.
4. In the **Client Time-out (secs)** text box, type the time-out value (for example, **100**).
5. Click **OK**.

DataStream

The NetScaler® DataStream™ feature provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a decision based on an SQL query parameters. You can further configure monitors to track the state of database servers.

Note: NetScaler DataStream is supported only for MySQL and MS SQL databases. For information about the supported protocol version, character sets, special queries, and transactions, see the Appendix NetScaler DataStream Reference.

How NetScaler DataStream Works

In DataStream, the NetScaler is placed in-line between the application and/or Web servers and the database servers. On the NetScaler appliance, the database servers are represented by services.

A typical DataStream deployment consists of the entities described in the following diagram.

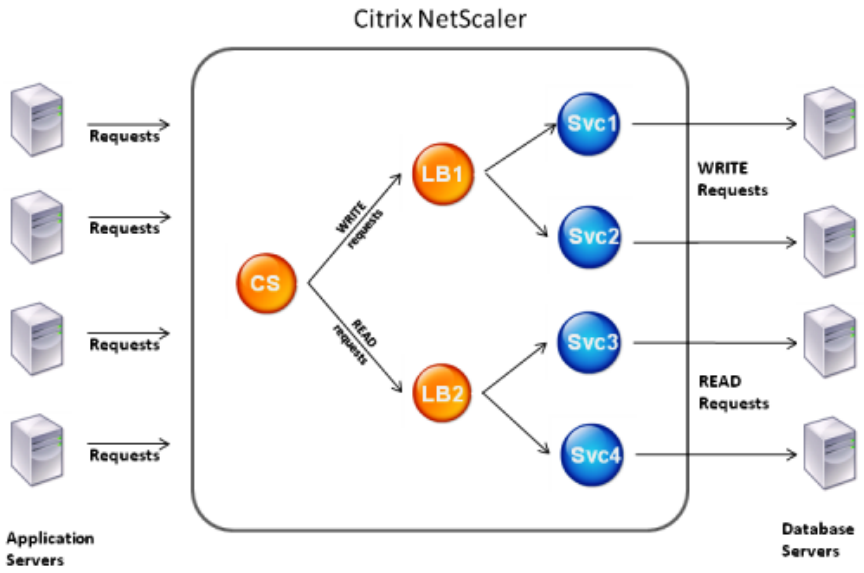


Figure 1. *DataStream Entity Model*

As shown in this figure, a DataStream configuration can consist of an optional content switching virtual server (CS), a load balancing setup consisting of load balancing virtual servers (LB1 and LB2) and services (Svc1, Svc2, Svc3, and Svc4), and content switching policies (optional).

The clients (application or Web servers) send requests to the IP address of a content switching virtual server (CS) configured on the NetScaler appliance. The NetScaler, then, authenticates the clients using the database user credentials configured on the NetScaler appliance. The content switching virtual server (CS) applies the associated content switching policies to the requests. After evaluating the policies, the content switching virtual server (CS) routes the requests to the appropriate load balancing virtual server (LB1 or LB2), which, then, distributes the requests to the appropriate database servers (represented by services on the NetScaler) based on the load balancing algorithm. The NetScaler uses the same database user credentials to authenticate the connection with the

database server.

If a content switching virtual server is *not* configured on the NetScaler, the clients (application or Web servers) send their requests to the IP address of a load balancing virtual server configured on the NetScaler appliance. The NetScaler authenticates the client by using the database user credentials configured on the NetScaler appliance, and then uses the same credentials to authenticate the connection with the database server. The load balancing virtual server distributes the requests to the database servers according to the load balancing algorithm. The most effective load balancing algorithm for database switching is the least connection method.

DataStream uses connection multiplexing to enable multiple client-side requests to be made over the same server-side connection. The following connection properties are considered:

- User name
- Database name
- Packet size
- Character set

How NetScaler DataStream Works

In DataStream, the NetScaler is placed in-line between the application and/or Web servers and the database servers. On the NetScaler appliance, the database servers are represented by services.

A typical DataStream deployment consists of the entities described in the following diagram.

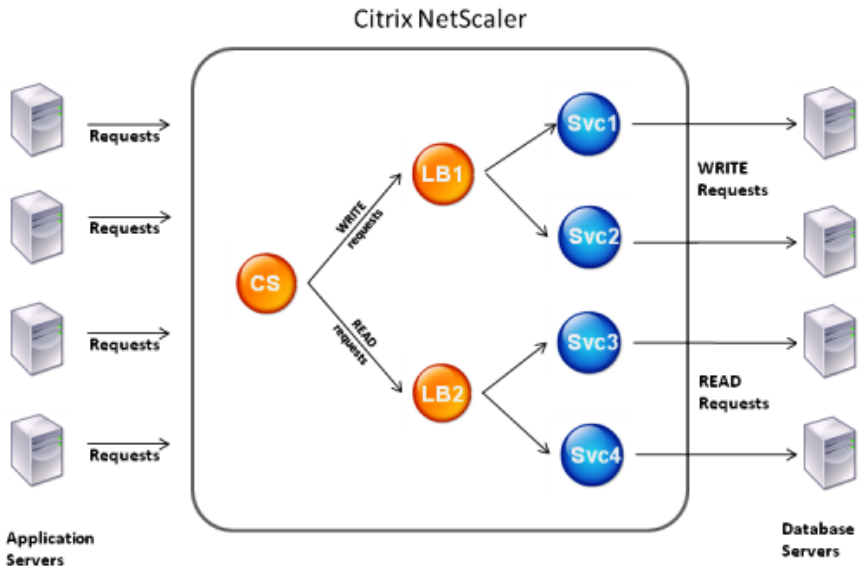


Figure 1. *DataStream Entity Model*

As shown in this figure, a DataStream configuration can consist of an optional content switching virtual server (CS), a load balancing setup consisting of load balancing virtual servers (LB1 and LB2) and services (Svc1, Svc2, Svc3, and Svc4), and content switching policies (optional).

The clients (application or Web servers) send requests to the IP address of a content switching virtual server (CS) configured on the NetScaler appliance. The NetScaler, then, authenticates the clients using the database user credentials configured on the NetScaler appliance. The content switching virtual server (CS) applies the associated content switching policies to the requests. After evaluating the policies, the content switching virtual server (CS) routes the requests to the appropriate load balancing virtual server (LB1 or LB2), which, then, distributes the requests to the appropriate database servers (represented by services on the NetScaler) based on the load balancing algorithm. The NetScaler uses the same database user credentials to authenticate the connection with the

database server.

If a content switching virtual server is *not* configured on the NetScaler, the clients (application or Web servers) send their requests to the IP address of a load balancing virtual server configured on the NetScaler appliance. The NetScaler authenticates the client by using the database user credentials configured on the NetScaler appliance, and then uses the same credentials to authenticate the connection with the database server. The load balancing virtual server distributes the requests to the database servers according to the load balancing algorithm. The most effective load balancing algorithm for database switching is the least connection method.

DataStream uses connection multiplexing to enable multiple client-side requests to be made over the same server-side connection. The following connection properties are considered:

- User name
- Database name
- Packet size
- Character set

Configuring Database Users

In databases, a connection is always stateful, which means that as soon as a connection is established, it must be authenticated.

You need to configure your database user name and password on the NetScaler. For example, if you have a user John configured on the database, you need to configure the user John on the NetScaler too. When you add the database user names and passwords on the NetScaler, these are added to the nsconfig file.

NetScaler uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

To add a database user by using the NetScaler command line

At the NetScaler command prompt, type

add db user <username> **-password** <password>

Example

```
> add db user nsdbuser -password dd260427edf
```

Parameters for creating a database user

username

The database user name. This is a mandatory argument. The maximum length of the user name is 127 characters.

password

The password used to log on to the database. The maximum length of the password is 127 characters.

To add a database user by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Database Users**.
2. In the details pane, click **Add**.
3. In the **Create Database User** dialog box, specify values for the following parameters.
   - **User Name**
   - **Password**
   - **Confirm Password**
4. Click **Create**, and then click **Close**. The service you created appears in the **Database Users** pane.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the NetScaler.

To reset the password of a database user by using the NetScaler command line

At the NetScaler command prompt, type

set db user <username> **-password** <password>

Example

```
> set db user nsdbuser -password dd260538abs
```

To reset the password of database users by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Database Users**.
2. In the details pane, select the database user for which you want to reset the password, and then click **Open**.
3. In the **Configure Database User** dialog box, modify the values for the following parameters.
   - **Password**
   - **Confirm Password**
4. Click **OK**.

If a database user no longer exists on the database server, you can remove the user from the NetScaler. However, if the user continues to exist on the database server and you remove the user from the NetScaler, any request from the client with this user name does not get authenticated, and therefore, does not get routed to the database server.

To remove a database user by using the NetScaler command line

At the NetScaler command prompt, type

rm db user <username>

Example

```
> rm db user nsdbuser
```

To remove a database user by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Database Users**.
2. In the details pane, select the database user that you want to remove, and then click **Open**.
3. In the **Proceed** message box, click **Yes**.

Configuring Load Balancing for DataStream

Before configuring a load balancing setup, you must enable the load balancing feature. Then, begin by creating at least one service for each database server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server and bind the services to the virtual server.

For instructions about configuring load balancing, see Load Balancing.

Parameter values specific to DataStream

Protocol

Use the MYSQL protocol type for MySQL databases and MS SQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see http://forge.mysql.com/wiki/MySQL_Internals_ClientServer_Protocol. For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

Port

Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

Method

It is recommended that you use the Least Connection method for better load balancing and lower server load. However, other methods, such asRound Robin, Least Response Time, Source IP Hash, Source IP Destination IP Hash, Least Bandwidth, Least Packets, and Source IP Source Port Hash, are also supported.

Note: URL Hash method is not supported for DataStream.

Configuring Content Switching for DataStream

You can segment traffic according to information in the SQL query, on the basis of database names, usernames, character sets, and packet size.

You can configure content switching policies with default syntax expressions to switch content based on connection properties, such as user name and database name, command parameters,and the SQL query to select the server.

The default syntax expressions evaluate traffic associated with MYSQL and MS SQL database servers. You can use request-based expressions in default syntax policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with MYSQL.RES)to evaluate server responses to user-configured health monitors.

Note: For information about default syntax expressions, see Default Syntax Expressions: DataStream.

Parameter values specific to DataStream

Protocol

Use the MYSQL protocol type for MySQL databases and MS SQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see http://forge.mysql.com/wiki/MySQL_Internals_ClientServer_Protocol. For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

Port

Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

For instructions about configuring content switching, see Content Switching.

Configuring Monitors for DataStream

To track the state of each load balanced database server in real time, you need to bind a monitor to each service. The monitor is configured to test the service by sending periodic probes to the service. (This is sometimes referred to as performing a health check.) If the monitor receives a timely response to its probes, it marks the service as UP. If it does not receive a timely response to the designated number of probes, it marks the service as DOWN.

For DataStream, you need to use the built-in monitosr, MYSQL-ECV and MSSQL-ECV. This monitor provides the ability to send an SQL request and parse the response for a string.

Before configuring monitors for DataStream, you must add database user credentials to your NetScaler. For information about configuring monitors, see Monitors.

When you create a monitor, a TCP connection is established with the database server, and the connection is authenticated by using the user name provided while creating the monitor. You can then run an SQL query to the database server and evaluate the server response to check whether it matches the configured rule.

Parameters specific to DataStream

type

Type of monitor. Use the MYSQL-ECV monitor type for MySQL databases and MSSQL-ECV monitor type for MS SQL databases.

sqlQuery

SQL query for the MYSQL-ECV and MSSQL-ECV monitors. After the connection to the database server is authenticated, you can run this query to the server.

evalRule

Rule evaluated to determine the state of the monitor. A rule consists of default syntax expressions.

database

Name of the database that needs to be probed. During authentication, this database name is used to connect to the database.

userName

User name to connect to the database. This is looked up in the database user list to extract the database.

Examples

In the following example, the value of the error message is evaluated to determine the state of the server.

```
add lb monitor lb_mon1 MYSQL_ECV -sqlQuery "select * from
table2;" -evalrule "mysql.res.error.message.contains(\"Invalid
User\")"-database "NS" -userName "user1"
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
add lb monitor lb_mon4 MYSQL_ECV -sqlQuery "select * from
table4;" -evalrule "mysql.res.atleast_rows_count(7)" -database "NS" -userName "user2"
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

```
add lb monitor lb_mon3 MYSQL_ECV
-sqlQuery "select * from ABC;" -evalrule "mysql.res.row(1).double_elem(2) == 345.12"
-database "NS" -userName "user3"
```

Use Case

A commonly used deployment scenario is the master/slave database architecture where the master database replicates all information to the slave databases.

For master/slave database architecture, you may want all WRITE requests to be sent to the master database and all READ requests to the slave databases.

The following figure shows the entities and the values of the parameters you need to configure on the appliance.

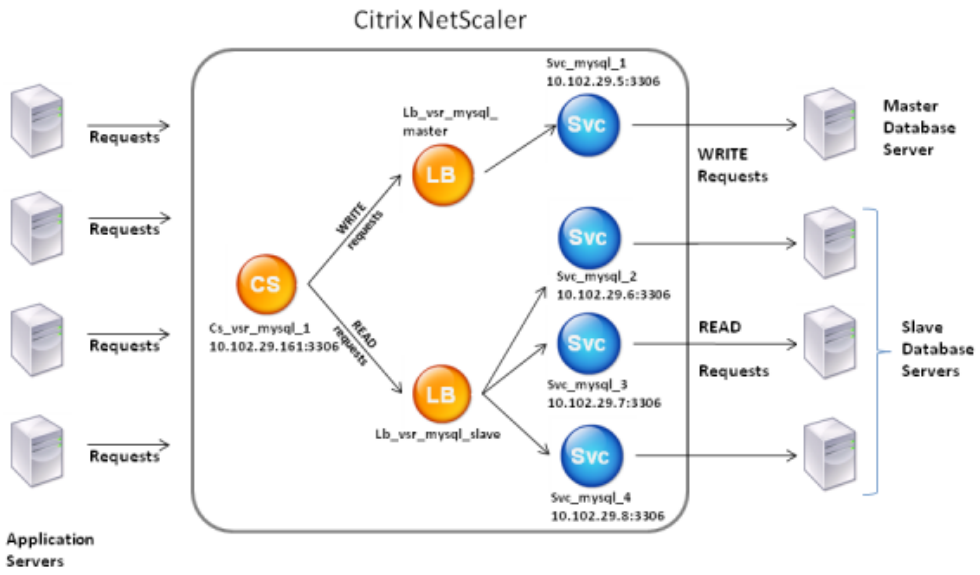


Figure 1. *DataStream Entity Model for Master/Slave Database Setup*

In this example scenario, a service (Svc_mysql_1) is created to represent the master database and is bound to a load balancing virtual server (Lb_vsr_mysql_master). Three more services (Svc_mysql_2, Svc_mysql_3, and Svc_mysql_4) are created to represent the three slave databases, and they are bound to another load balancing virtual server (Lb_vsr_mysql_slave).

A content switching virtual server (Cs_vsr_mysql_1) is configured with associated policies to send all WRITE requests to the load balancing virtual server, Lb_vsr_mysql_master, and all READ requests to the load balancing virtual server, Lb_vsr_mysql_slave.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. After evaluating the policies, the

content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

The following table lists the names and values of the entities and the policy configured on the NetScaler.

Table 1. *Entity and Policy Names and Values*

Entity Type	Name	IP Address	Protocol	Port	Expression
Services	Svc_mysql_1	10.102.29.5	MYSQL	3306	NA
	Svc_mysql_2	10.102.29.6	MYSQL	3306	NA
	Svc_mysql_3	10.102.29.7	MYSQL	3306	NA
	Svc_mysql_4	10.102.29.8	MYSQL	3306	NA
Load balancing virtual servers	Lb_vsr_mysql_master	10.102.29.201	MYSQL	3306	NA
	Lb_vsr_mysql_slave	10.102.29.202	MYSQL	3306	NA
Content switching virtual server	Cs_vsr_mysql_1	10.102.29.161	MYSQL	3306	NA
Content switching policy	Cs_select	NA	NA	NA	"MYSQL.REQ.QUERY.COMMAND.contains(\"selec

To configure DataStream for a master/slave database setup by using the NetScaler command line

At the NetScaler command prompt, type

- add service Svc_mysql_1 10.102.29.5 mysql 3306
- add service Svc_mysql_2 10.102.29.6 mysql 3306
- add service Svc_mysql_3 10.102.29.7 mysql 3306
- add service Svc_mysql_4 10.102.29.8 mysql 3306
- add lb vserver Lb_vsr_mysql_master mysql 10.102.29.201 3306
- add lb vserver Lb_vsr_mysql_slave mysql 10.102.29.202 3306
- bind lb vserver Lb_vsr_mysql_master svc_mysql_1
- bind lb vserver Lb_vsr_mysql_slave svc_mysql_2

- bind lb vserver Lb_vsr_mysql_slave svc_mysql_3
- bind lb vserver Lb_vsr_mysql_slave svc_mysql_4
- add cs vserver Cs_vsr_mysql_1 mysql 10.102.29.161 3306
- add cs policy Cs_select –rule "MYSQL.REQ.QUERY.COMMAND.contains(\"select\")"
- bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_master
- bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_slave –policy Cs_select –priority 10

To configure DataStream for a master/slave database setup by using the configuration utility

Add four services, one to represent the master database server and three to represent the slave database servers.

To add a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters as listed in Entity and Policy Names and Values.
   - Service Name
   - IP Address
   - Protocol
   - Port
4. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Add two load balancing virtual servers.

To create a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters as listed in Entity and Policy Names and Values.
   - Name
   - IP Address
   - Protocol
   - Port
4. Click **Create**, and then click **Close**.

Bind the service Svc_mysql_1 to the load balancing virtual server Lb_vsr_mysql_master, and bind the three services (Svc_mysql_2, Svc_mysql_3, and Svc_mysql_4) to the load balancing virtual server Lb_vsr_mysql_slave.

To bind a service to a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind the service (for example, **Lb_vsr_mysql_master**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Svc_mysql_1**).
5. Click **OK**.

Create a content switching virtual server.

To add a content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Content Switching)** dialog box, specify values for the following parameters as listed in Entity and Policy Names and Values.
   - Name
   - IP Address
   - Protocol
   - Port
4. Click **Create**, and then click **Close**.

Create a content switching policy to evaluate all READ requests.

To create a content switching policy by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, click **Add**.
3. In the **Create Content Switching Policy** dialog box, in the **Name** text box, type the name of the policy (for example, **Cs_select**).
4. Choose the type of policy that you want to create, and configure the policy. To create a rule-based policy, click **Configure**, and do the following:
   - In the **Create Expression** dialog box, choose the expression syntax you want to use and enter your policy expressions as listed in Entity and Policy Names and Values.
5. Click **Create**, and then click **Close**.

Bind the content switching policy to the content switching virtual server. You should also select a load balancing virtual server as the target for the policy so that, after the content switching virtual server evaluates the policy, it routes requests that match the policy to the load balancing virtual server to forward them to the appropriate database server.

To bind the policy to the content switching virtual server and select a load balancing virtual server target by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, double-click the virtual server to which you want to bind the policy (for example, **Cs_vsr_mysql_1**).
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Policies** tab, click **Insert Policy**, and in the **Policy Name** column, select the policy that you want to bind to the virtual server (for example, **Cs_select**).
4. In the **Target** column next to the policy, select the load balancing virtual server that you want to assign as the target for the policy (for example, **Lb_vsr_mysql_slave**).
5. Click **OK**.

Set the load balancing virtual server, Lb_vsr_mysql_master, as the default virtual server for the content switching virtual server by binding the content switching virtual server to this load balancing virtual server. This ensures that the content switching virtual server routes requests that do not match the Cs_select policy to the load balancing virtual server to forward them to the appropriate database server.

To bind the content switching virtual server to a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In the details pane, double-click the virtual server to which you want to bind the policy (for example, **Cs_vsr_mysql_1**).
3. In the **Configure Virtual Server (Content Switching)** dialog box, on the **Policies** tab, click **Insert Policy**, and in the **Policy Name** column, select **(Default)**.
4. In the **Target** column next to the policy, select the load balancing virtual server that you want to assign as the target for the policy (for example, **Lb_vsr_mysql_master**).
5. Click **OK**.

Flex Tenancy

Flex Tenancy™ is a Citrix NetScaler methodology that allows you to tune a group of NetScaler VPX instances to the unique characteristics and needs of individual applications in a complex Web 2.0 setup.

As the Web becomes the de-facto way of delivering application services, the number of web applications that enterprises and service providers must support has been growing at a very fast pace. Not only are the technical requirements of each web application different, but also in most large environments, the applications are owned and controlled by different constituencies/organizations. This leads to the need to offer the Application Delivery Controller (ADC) functionality as a shared service to multiple tenants.

Understanding the Flex Tenancy Architecture

In a typical Flex Tenancy set-up, NetScaler physical and virtual appliances can be used together in a two-tier architecture with each tier dedicated to managing specific actions. This:

- provides flexibility for how application delivery services are deployed and managed
- makes it much easier to shuffle applications and their associated application delivery resources on-demand without risking the stability of the entire network
- reduces costs

The Flex Tenancy architecture segments web application delivery services into two tiers:

- shared-network-service flex tier
- application-specific tenant tier

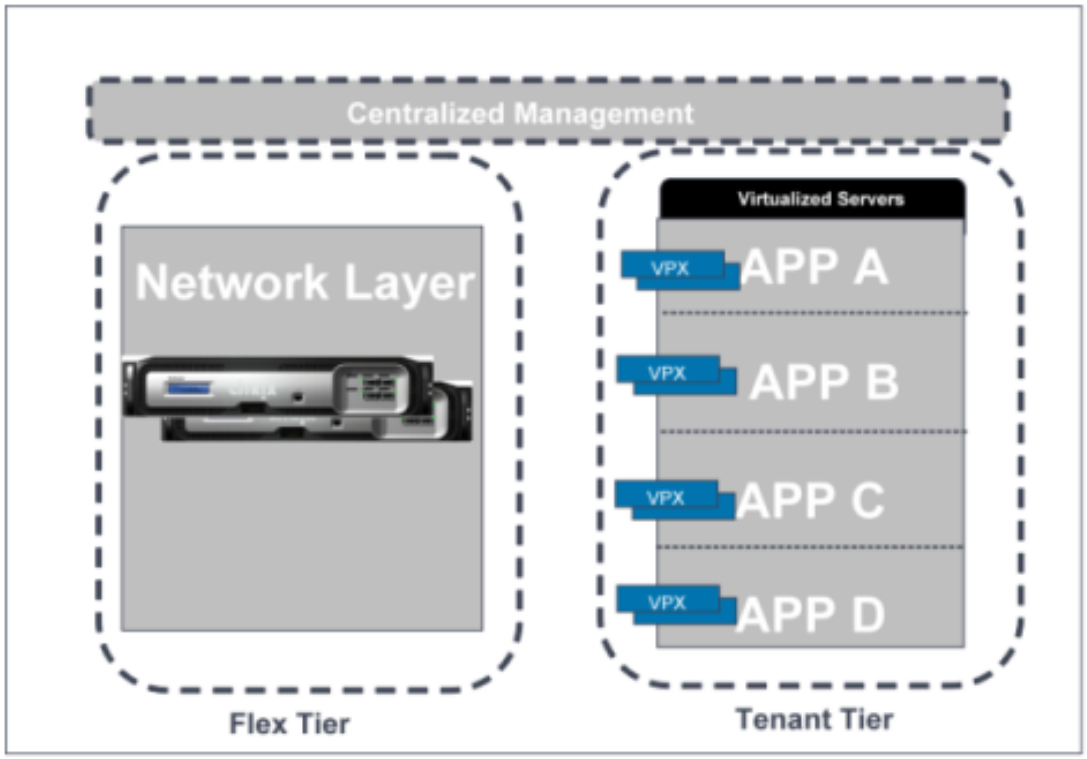


Figure 1. Flex Tenancy Architecture

The Flex Tier

The flex tier runs at the edge of the data center or an enterprise network. The flex tier generally performs application delivery services and associated policies that are common and applied to all applications. Examples of such services could include:

- SSL offload
- SSL VPN
- DoS mitigation
- GSLB
- Certain types of redirects
- Content Switching

Since the policies are common and shared, there is no reason to try and partition them across multiple NetScaler instances dedicated per specific application. Indeed, partitioning at the flex tier is counterproductive as it leads to duplication of effort and the associated possibility of configuration errors during change management. Therefore, the flex tier should be physically centralized on as few NetScaler instances as possible. In all but the smallest environments this will generally lead to NetScaler MPX physical appliances being used at the flex tier.

The Tenant Tier

The tenant tier runs logically close to the applications, with each tenant (whether a tenant is an application, a line of business, or a customer) getting its own NetScaler instance. The tenant tier isolates the application delivery needs that vary by application - like server load balancing, caching, compression and application firewall protection - per application.

Since the policies - and thus the NetScaler configurations - differ by application, there is no inherent configuration benefit to centralizing multiple applications onto a single NetScaler. In fact, there are significant configuration isolation, separation of duties and capacity planning benefits of dedicating appliances on an app-by-app basis.

Centralized Management

One potential downside to dedicating NetScaler instances to specific applications is "appliance sprawl". In Flex Tenancy architectures, the vast majority of most instances will be NetScaler VPX virtual appliances. Therefore, the physical challenges (e.g., racking, cabling) associated with appliance sprawl don't exist. Citrix Command Center is used to provide centralized FCAPS (fault, configuration, accounting, performance, security) management of both the flex and the tenant tiers.

Understanding the Flex Tenancy Architecture

In a typical Flex Tenancy set-up, NetScaler physical and virtual appliances can be used together in a two-tier architecture with each tier dedicated to managing specific actions. This:

- provides flexibility for how application delivery services are deployed and managed
- makes it much easier to shuffle applications and their associated application delivery resources on-demand without risking the stability of the entire network
- reduces costs

The Flex Tenancy architecture segments web application delivery services into two tiers:

- shared-network-service flex tier
- application-specific tenant tier

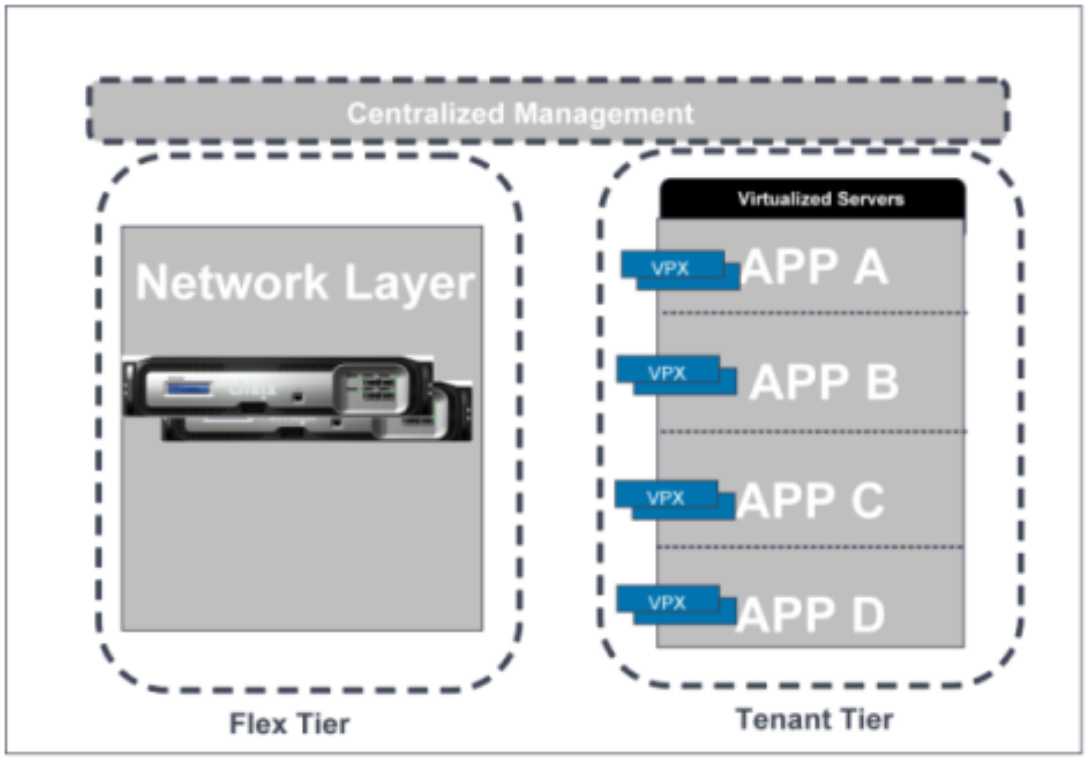


Figure 1. Flex Tenancy Architecture

The Flex Tier

The flex tier runs at the edge of the data center or an enterprise network. The flex tier generally performs application delivery services and associated policies that are common and applied to all applications. Examples of such services could include:

- SSL offload
- SSL VPN
- DoS mitigation
- GSLB
- Certain types of redirects
- Content Switching

Since the policies are common and shared, there is no reason to try and partition them across multiple NetScaler instances dedicated per specific application. Indeed, partitioning at the flex tier is counterproductive as it leads to duplication of effort and the associated possibility of configuration errors during change management. Therefore, the flex tier should be physically centralized on as few NetScaler instances as possible. In all but the smallest environments this will generally lead to NetScaler MPX physical appliances being used at the flex tier.

The Tenant Tier

The tenant tier runs logically close to the applications, with each tenant (whether a tenant is an application, a line of business, or a customer) getting its own NetScaler instance. The tenant tier isolates the application delivery needs that vary by application - like server load balancing, caching, compression and application firewall protection - per application.

Since the policies - and thus the NetScaler configurations - differ by application, there is no inherent configuration benefit to centralizing multiple applications onto a single NetScaler. In fact, there are significant configuration isolation, separation of duties and capacity planning benefits of dedicating appliances on an app-by-app basis.

Centralized Management

One potential downside to dedicating NetScaler instances to specific applications is "appliance sprawl". In Flex Tenancy architectures, the vast majority of most instances will be NetScaler VPX virtual appliances. Therefore, the physical challenges (e.g., racking, cabling) associated with appliance sprawl don't exist. Citrix Command Center is used to provide centralized FCAPS (fault, configuration, accounting, performance, security) management of both the flex and the tenant tiers.

Building a Flex Tenancy Solution

The Flex Tenancy architecture is applicable to any high volume environment where:

- multiple tenants share a common infrastructure
- there is a core set of shared application delivery services that are common to all tenants
- each tenant also has its own specific web application delivery needs and requirements

The two most prevalent examples of this kind of environment are:

- hosting/managed services providers that want to offer dedicated load balancing/application delivery services to their customers
- an enterprise IT organization that models itself as an “internal service provider” to support multiple lines-of-business and business applications.

In each case, the core architecture is similar. However, the deployment architecture and the management of the overall system will likely vary based upon the very different business needs of each case.

Enterprise IT as an Internal Service Provider

Many enterprise IT organizations model themselves as internal service providers supporting the different needs of the different lines-of-business. Building "private clouds" to support enterprise apps is this the latest phase of this trend.

An enterprise typically hosts a variety of applications that may be independent of each other.

These applications:

- require their own application server pools
- typically need customized ADC policies and tuning

  have their own distinct change management requirements and windows

The challenge facing the groups managing the web application delivery infrastructure is to support different and increasingly complex applications and at the same time scale so that they can support an ever increasing number of applications.

Achieving application performance in this environment depends upon coordination between network teams and the various application teams. Application tuning requires knowledge of the application and in many cases should be responsibility of the application team. Maintaining network performance and availability requires knowledge of various networking protocols and hardware management and therefore becomes the responsibility of the network team.

Sharing the same device between both teams creates manageability challenges for an organization. For example, consider an application administrator wanting to upgrade to upgrade the ADC take advantage of new functionality. If the ADC is shared by other applications, the administrator will must coordinate with all these teams to initiate change control for his change.

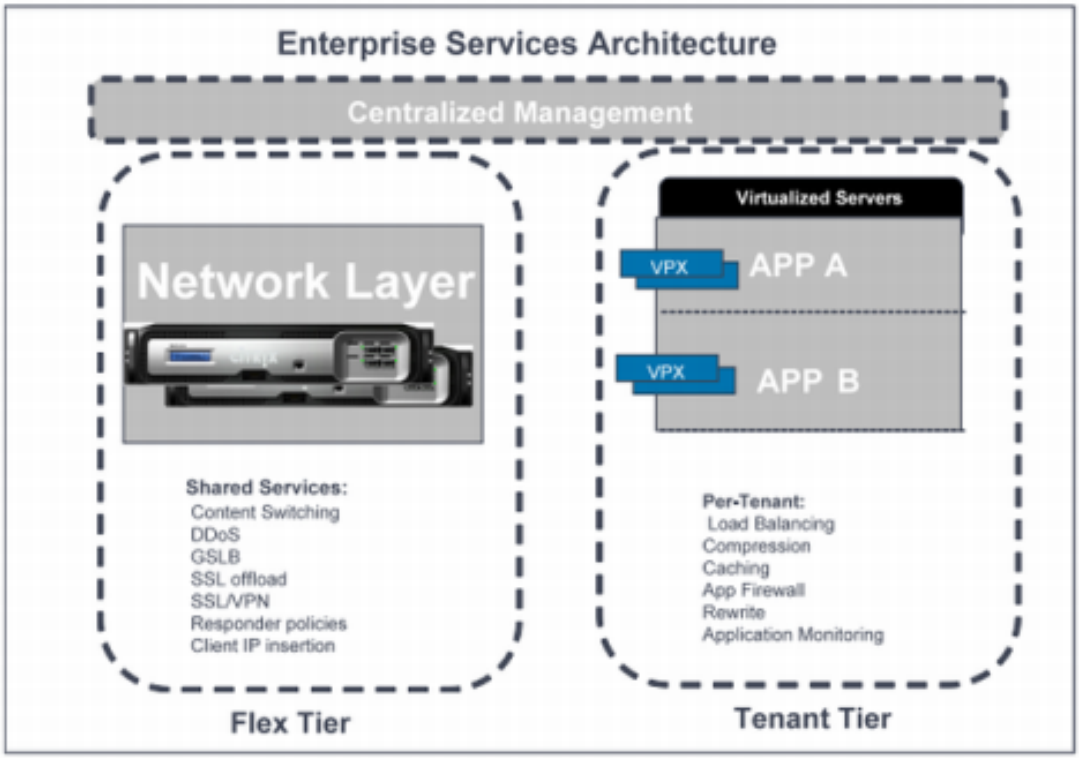
Moreover, any changes to the ADC will need to be tested with every other application. As the number of applications grows, this model becomes exponentially complex.

The Flex Tenancy architecture addresses these issues. Since no other applications run on the tenant instance:

- it can be upgraded based solely on the needs of its hosted app
- there is no need to qualify the version upgrade against other applications

  there is no concern that using the new functionality will impact other app performance

Figure 1. Architecture of an enterprise service solution



There are three possible administrative approaches using Flex Tenancy:

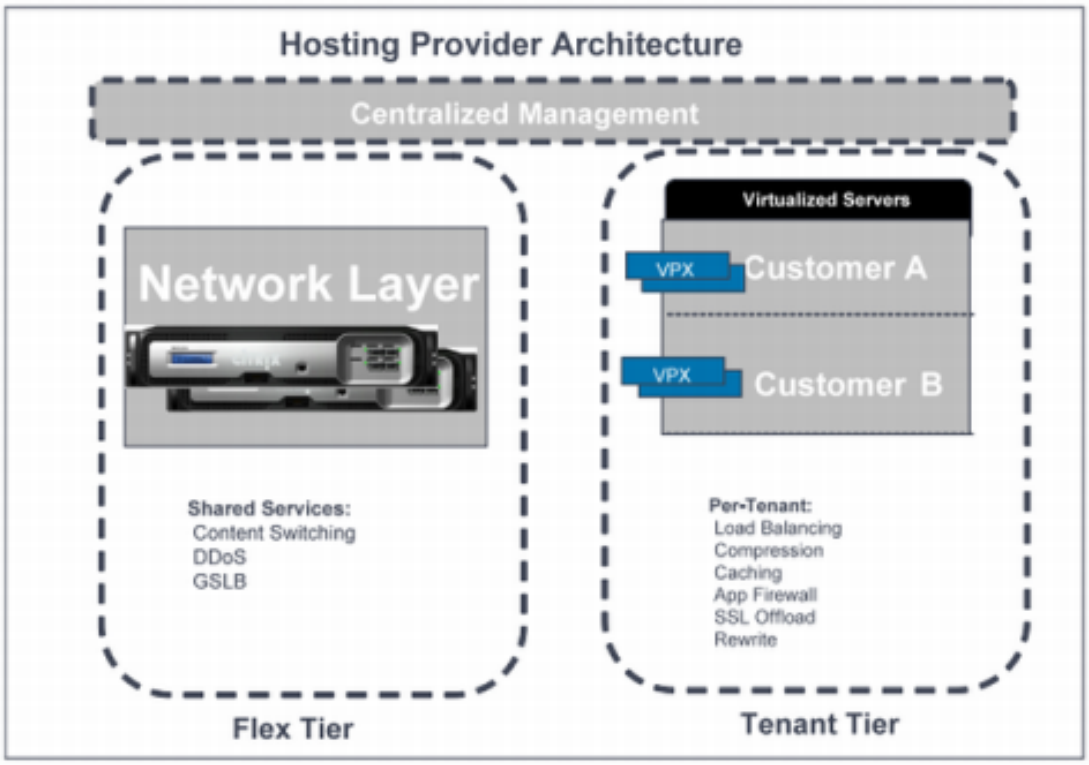
1. The central IT organization can retain full administrative control over both the flex and tenant tiers.

2. The central IT organization can retain full control over the flex tier, but delegate management rights of the tenant instances to individual app teams.

3. The central IT team can retain full control over the flex tier, retain control over certain elements of the tenant tier (e.g., certain IP addressing, ensuring it can retain “root access”) but delegate control of other functionality to app teams.

The ability to tune the application delivery fabric to individual applications enables IT organizations to maximize the performance of their applications and to scale the number of applications they provide to their stakeholders without significant increase in costs.

Hosting provider solution



A hosting provider provides space on a server they own and Internet connectivity for the applications hosted on that server. These servers are typically hosted in a data center. Load balancing is a frequently requested "add-on" services customers ask for. Additionally, advanced L7 services such as compression, application firewall and caching provide additional revenue opportunities for the hosting provider.Figure 1. Architecture of a hosting provider solution

The biggest challenge in this environment is managing and delivering application delivery services for a wide variety of customers, each with varying requirements. Trying to manage these services on a shared device can lead to performance bottlenecks, change management conflicts and ultimately a loss of business. Also, many customers will want admin rights to control the load balancing services used for their hosted back-end servers.

Flex Tenancy solution can be very useful in this scenario. A hosting provider can use the flex tier to run services common to the entire data center. These include global server load balancing (GSLB) which would provide data center redundancy, denial of service protection (DoS), and segregate traffic to the tenant tier using content switching policies.

At the tenant tier each tenant is provisioned, and can manage its own, dedicated NetScaler instances.

One of the primary services of this tier is to load balance requests to the backend servers. In addition, this tier may also perform SSL offload. This is usually a flex tier service, but in a hosting environment it may be offloaded to the tenant tier if the hosting business model

supports customers owning and managing their own certificates.

Tenant tier is also used to apply a variety of complex application policies to tune application performance. For example, a customer can define caching policies to cache objects of a certain application only for ten seconds while other application objects could be cached for a longer duration. Customers can also define rewrite policies, application firewall policies, compression policies and application level monitoring policies.

In this environment, flex tier can comprise of one or more MPX devices while the tenant tier could comprise of a combination of VPX and MPX devices. Citrix provides Command Center to manage these devices.

Since VPX is a virtual appliance, network administrators can easily create a workflow, using commercial or in-house data center automation tools, to bring up additional NetScaler VPX instances and servers online to support the increased load. NetScaler also provides extensive API that can be used to integrate into the work flow automation tools used in hosting/managed services infrastructures. Network administrators can also use the Citrix Command Center in conjunction to propagate the relevant NetScaler configurations to these newly started NetScaler VPX instances.

The division of services amongst the tiers and the ability to manage them through Command Center and NetScaler APIs enables a hosting provider to scale performance of his application delivery fabric and to streamline his operations.

HTTP Callouts

For certain types of requests, or when certain criteria are met during policy evaluation, you might want to stall policy evaluation briefly, retrieve information from a server, and then perform a specific action that depends on the information that is retrieved. At other times, when you receive certain types of requests, you might want to update a database or the content hosted on a Web server. HTTP callouts enable you to perform all these tasks.

An HTTP callout is an HTTP request that the NetScaler appliance generates and sends to an external application when certain criteria are met during policy evaluation. The information that is retrieved from the server can be analyzed by default syntax policy expressions, and an appropriate action can be performed. You can configure HTTP callouts for HTTP content switching, TCP content switching, rewrite, responder, and for the token-based method of load balancing.

Before you configure an HTTP callout, you must set up an application on the server to which the callout will be sent. The application, which is called the *HTTP callout agent* , must be configured to respond to the HTTP callout request with the required information. The HTTP callout agent can also be a Web server that serves the data for which the NetScaler appliance sends the callout. You must make sure that the format of the response to an HTTP callout does not change from one invocation to another.

After you set up the HTTP callout agent, you configure the HTTP callout on the NetScaler appliance. Finally, to invoke the callout, you include the callout in a default syntax policy in the appropriate NetScaler feature and then bind the policy to the bind point at which you want the policy to be evaluated.

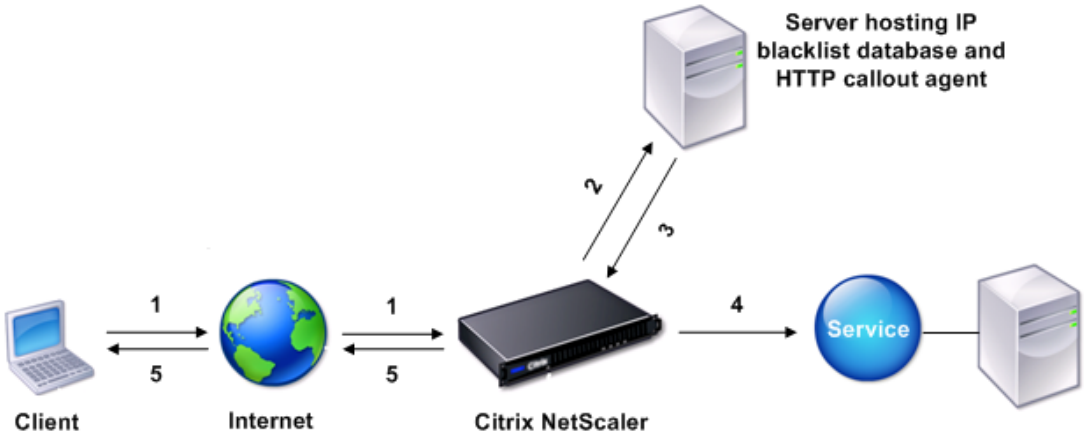
After you have configured the HTTP callout, you must verify the configuration to make sure that the callout is working correctly.

How an HTTP Callout Works

When the NetScaler appliance receives a client request, the appliance evaluates the request against the policies bound to various bind points. During this evaluation, if the appliance encounters the HTTP callout expression, SYS.HTTP_CALLOUT(<name>), it stalls policy evaluation briefly and sends an HTTP request to the HTTP callout agent by using the parameters configured for the specified HTTP callout. Upon receiving the response, the appliance inspects the specified portion of the response, and then either performs an action or evaluates the next policy, depending on whether the evaluation of the response from the HTTP callout agent evaluates to TRUE or FALSE, respectively. For example, if the HTTP callout is included in a responder policy, if the evaluation of the response evaluates to TRUE, the appliance performs the action associated with the responder policy.

If the HTTP callout configuration is incorrect or incomplete, or if the callout invokes itself recursively, the appliance raises an UNDEF condition, and updates the undefined hits counter.

The following figure illustrates the working of an HTTP callout that is invoked from a globally bound responder policy. The HTTP callout is configured to include the IP address of the client that is associated with an incoming request. When the NetScaler appliance receives a request from a client, the appliance generates the HTTP callout request and sends it to the callout server, which hosts a database of blacklisted IP addresses and an HTTP callout agent that checks whether the client's IP address is listed in the database. The HTTP callout agent receives the HTTP callout request, checks whether the client's IP address is listed, and sends a response that the NetScaler appliance evaluates. If the response indicates that the client's IP address is not blacklisted, the appliance forwards the response to the configured service. If the client's IP address is blacklisted, the appliance resets the client connection



1: Client request
2: HTTP callout request to check whether the client is blacklisted
3: Response from HTTP callout agent
4: Request forwarded to service if 3 indicates a safe IP address
5: Connection RESET if 3 indicates a bad IP address

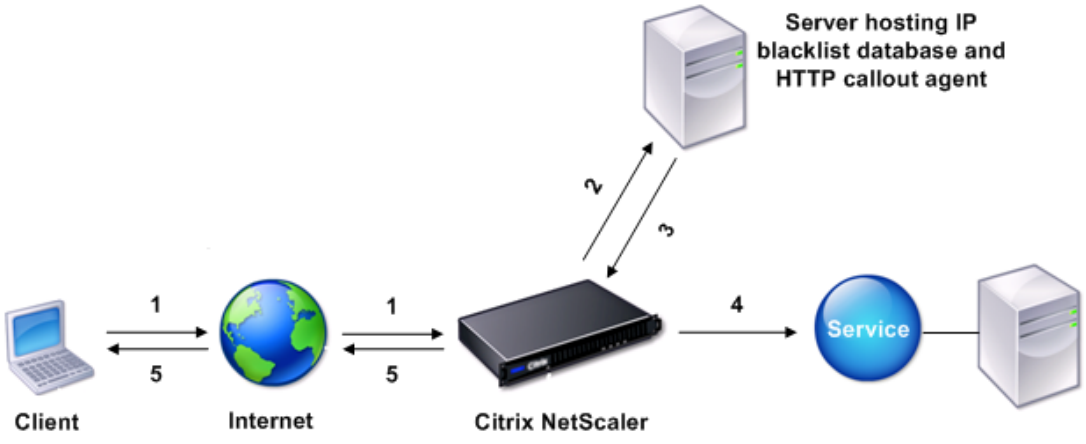
Figure 1. HTTP Callout Entity Model

How an HTTP Callout Works

When the NetScaler appliance receives a client request, the appliance evaluates the request against the policies bound to various bind points. During this evaluation, if the appliance encounters the HTTP callout expression, SYS.HTTP_CALLOUT(<name>), it stalls policy evaluation briefly and sends an HTTP request to the HTTP callout agent by using the parameters configured for the specified HTTP callout. Upon receiving the response, the appliance inspects the specified portion of the response, and then either performs an action or evaluates the next policy, depending on whether the evaluation of the response from the HTTP callout agent evaluates to TRUE or FALSE, respectively. For example, if the HTTP callout is included in a responder policy, if the evaluation of the response evaluates to TRUE, the appliance performs the action associated with the responder policy.

If the HTTP callout configuration is incorrect or incomplete, or if the callout invokes itself recursively, the appliance raises an UNDEF condition, and updates the undefined hits counter.

The following figure illustrates the working of an HTTP callout that is invoked from a globally bound responder policy. The HTTP callout is configured to include the IP address of the client that is associated with an incoming request. When the NetScaler appliance receives a request from a client, the appliance generates the HTTP callout request and sends it to the callout server, which hosts a database of blacklisted IP addresses and an HTTP callout agent that checks whether the client's IP address is listed in the database. The HTTP callout agent receives the HTTP callout request, checks whether the client's IP address is listed, and sends a response that the NetScaler appliance evaluates. If the response indicates that the client's IP address is not blacklisted, the appliance forwards the response to the configured service. If the client's IP address is blacklisted, the appliance resets the client connection



1: Client request
2: HTTP callout request to check whether the client is blacklisted
3: Response from HTTP callout agent
4: Request forwarded to service if 3 indicates a safe IP address
5: Connection RESET if 3 indicates a bad IP address

Figure 1. HTTP Callout Entity Model

Notes on the Format of HTTP Requests and Responses

The NetScaler appliance does not check for the validity of the HTTP callout request. Therefore, before you configure HTTP callouts, you must know the format of an HTTP request. You must also know the format of an HTTP response, because configuring an HTTP callout involves configuring expressions that evaluate the response from the HTTP callout agent.

Format of an HTTP Request

An HTTP request contains a series of lines that each end with a carriage return and a line feed, represented as either <CR><LF> or \r\n.

The first line of a request (the *message line*) contains the HTTP method and target. For example, a message line for a GET request contains the keyword GET and a string that represents the object that is to be fetched, as shown in the following example:

GET /mysite/mydirectory/index.html HTTP/1.1\r\n

The rest of the request contains HTTP headers, including a required Host header and, if applicable, a message body.

The request ends with a bank line (an extra <CR><LF> or \r\n).

Following is an example of a request:

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: */*\r\n
\r\n
```

Format of an HTTP Response

An HTTP response contains a status message, response HTTP headers, and the requested object or, if the requested object cannot be served, an error message.

Following is an example of a response:

```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag: “04f97692cbd1:377”\r\n
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
<55-character response>
```

Configuring an HTTP Callout

When configuring an HTTP callout, you specify the destination and format of the HTTP request, the expected format of the response, and, finally, the portion of the response that you want to analyze.

For the destination, you either specify the IP address and port of the HTTP callout agent or engage a load balancing, content switching, or cache redirection virtual server to manage the HTTP callout requests. In the first case, the HTTP callout requests will be sent directly to the HTTP callout agent. In the second case, the HTTP callout requests will be sent to the virtual IP address (VIP) of the specified virtual server. The virtual server will then process the request in the same way as it processes a client request. For example, if you expect a large number of callouts to be generated, you can configure instances of the HTTP callout agent on multiple servers, bind these instances (as services) to a load balancing virtual server, and then specify the load balancing virtual server in the HTTP callout configuration. The load balancing virtual server then balances the load on those configured instances as determined by the load balancing algorithm.

For the format of the HTTP callout request, you can specify the individual attributes of the HTTP callout request (an attribute-based HTTP callout), or you can specify the entire HTTP callout request as a default syntax expression (an expression-based HTTP callout).

Note: The NetScaler appliance does not check for the validity of the HTTP request. You must make sure that the HTTP request is a valid request. An incorrect or incomplete HTTP callout configuration results in a runtime UNDEF condition that is not associated with an action. The UNDEF condition merely updates the Undefined Hits counter, which enables you to troubleshoot an incorrectly configured HTTP callout. However, the appliance parses the HTTP callout request to enable you to configure certain NetScaler features for the callout. This can lead to an HTTP callout invoking itself. For information about callout recursion and how you can avoid it, see Avoiding HTTP Callout Recursion.

Finally, regardless of whether you use HTTP request attributes or an expression to define the format of the HTTP callout request, you must specify the format of the response from the HTTP callout agent and the portion of the response that you want to evaluate. The response can be a Boolean value, a number, or text. The portion of the response that you want to evaluate is specified by an expression. For example, if you specify that the response contains text, you can use `HTTP.REQ.BODY(<uint>)` to specify that the appliance must evaluate only the first <uint> bytes of the response from the callout agent.

At the NetScaler command line, you first create an HTTP callout by using the add command. When you add a callout, all parameters are set to a default value of NONE, except the HTTP method, which is set to a default value of GET. You then configure the callout's parameters by using the set command. The set command is used to configure both types of callouts (attribute-based and expression-based). The difference lies in the parameters that are used for configuring the two types of callouts. Accordingly, the command-line instructions that follow include a set command for configuring an attribute-based callout and a set command for configuring an expression-based callout. In the NetScaler configuration utility, all of these configuration tasks are performed in a single dialog box.

Note: Before you put an HTTP callout into a policy, you can modify all configured parameters except the return type. Once an HTTP callout is in a policy, you cannot completely modify an expression that is configured in the callout. For example, you cannot change `HTTP.REQ.HEADER("myval")` to `CLIENT.IP.SRC`. However, you can modify the operators and arguments that are passed to the expression. For example, you can change `HTTP.REQ.HEADER("myVal1")` to `HTTP.REQ.HEADER("myVal2")`, or `HTTP.REQ.HEADER("myVal")` to `HTTP.REQ.HEADER("myVal").AFTER_STR(<string>)`. If the set command fails, create a new HTTP callout.

HTTP callout configuration involves configuring default syntax expressions. For more information about configuring default syntax expressions, see Configuring Default Syntax Expressions: Getting Started.

To create an HTTP callout by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an HTTP callout and verify the configuration:

- add policy httpCallout <name>
- show policy httpCallout <name>

Example

```
> add policy httpCallout mycallout
Done
> show policy httpCallout mycallout
    Name: mycallout
   Server: :0 (DOWN)
Return type: NONE
Method: GET
 Host expr: NONE
URL stem: NONE
Headers: NONE
 Parameters: NONE
Result expr: NONE
 Hits: 0
 Undef Hits: 0
Last undef reason: Config incomplete
Done
>
```

To configure an attribute-based HTTP callout by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an HTTP callout, configure the callout based on request attributes, and then verify the configuration:

- add policy httpCallout <name>
- set policy httpCallout <name> [-**IPAddress** <ip_addr|ipv6_addr|*>] [-**port** <port|*>] [-**vServer** <string>] [-**returnType** <returnType>] [-**httpMethod** (GET | POST)] [-**hostExpr** <string>] [-**urlStemExpr** <string>] [-**headers** <name(value)> ...] [-**parameters** <name(value)> ...] [-**resultExpr** <string>]
- show policy httpCallout <name>

Example

```
> add policy httpCallout mycallout
Done
 set policy httpCallout mycallout -vserver lbv1 -returnType num -httpMethod GET -hostExpr 'http.req.header
 Done
> show policy httpCallout mycallout
Name: mycallout
Vserver: lbv1 (UP)
Return type: NUM
Method: GET
 Host expr: http.req.header("Host")
 URL stem: http.req.url
 Headers: Name("MyHeader")
 Parameters: Name("My Name")
  Result expr: http.res.body(10000).length
 Hits: 0
 Undef Hits: 0
 Done
>
```

To configure an expression-based HTTP callout by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an HTTP callout, configure the callout based on an expression, and then verify the configuration:

- add policy httpCallout <name>
- set policy httpCallout <name> [-**vServer** <string>] [-**returnType** <returnType>] [-**httpMethod** (GET | POST)] [-**fullReqExpr** <string>] [-**resultExpr** <string>]
- show policy httpCallout <name>

Example

```
> add policy httpCallout mycallout1
 Done
 set policy httpCallout mycallout1 -vserver lbv1  -returnType num -httpMethod GET -fullReqExpr q{"GET " + h
 Done
 show policy httpCallout mycallout1
  Name: mycallout1
 Vserver: lbv1 (UP)
Return type: NUM
Full REQ expr: "GET " + http.req.url + "HTTP/" + http.req.version.major + "." + http.req.version.minor.sub(1)
Result expr: NONE
   Hits: 0
 Undef Hits: 0
Last undef reason: Config incomplete
Done
>
```

To set an HTTP callout parameter to the default value by using the NetScaler command line

At the NetScaler command prompt, type the following commands to view the configuration of an HTTP callout, set a parameter to the default value, and verify the configuration:

- show policy httpCallout <name>
- unset policy httpCallout <name> [-**IPAddress**] [-**port**] [-**vServer**] [-**httpMethod**] [-**hostExpr**] [-**urlStemExpr**] [-**headers**] [-**parameters**] [-**fullReqExpr**] [-**resultExpr**]<name>
- show policy httpCallout

Example

```
> show policy httpCallout myCallout
  Name: myCallout
        Server: 10.102.3.96:443 (DOWN)
        Return type: TEXT
        Method: POST
        Host expr: "10.102.3.95"
        URL stem: "/cgi-bin/check_clnt_from_database.pl"
        Headers: Request("Callout Request")
     Parameters: param1("user1") param2("user2")
        Result expr: HTTP.RES.BODY(100)
        Hits: 0
     Undef Hits: 0
     Last undef reason: Server is DOWN
 Done
> unset policy httpCallout myCallout -httpMethod
 Done
```

```
> show policy httpCallout myCallout
        Name: myCallout
        Server: 10.102.3.96:443 (DOWN)
        Return type: TEXT
                        Method: GET
        Host expr: "10.102.3.95"
        URL stem: "/cgi-bin/check_clnt_from_database.pl"
        Headers: Request("Callout Request")
        Parameters: param1("user1") param2("user2")
        Result expr: HTTP.RES.BODY(100)
        Hits: 0
        Undef Hits: 0
        Last undef reason: Server is DOWN
 Done
>
```

To modify or remove an HTTP callout by using the NetScaler command line

- To modify an HTTP callout, type the set policy httpCallout command, the name of the HTTP callout, and the parameters to be changed, with their new values.
- To remove an HTTP callout, type the rm policy httpCallout command and the name of the HTTP callout.

Parameters for configuring an HTTP callout

name

The name of the HTTP callout. This is a required argument. Maximum length: 31.

IPAddress

The IPv4 or IPv6 address of the server to which the callout is sent. Or, it can be a wildcard. Mutually exclusive with the `-vserver` parameter.

port

If you specify an IP address, this is the port on the server to which the HTTP callout agent is mapped. Or, it can be a wildcard. Mutually exclusive with the `-vserver` parameter.

vServer

The name of the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent. The service type of the virtual server must be `HTTP`.

This option is mutually exclusive with `-IPAddress` and `-port`. The maximum length is 127 characters.

returnType

Type of data that the target application returns in the response to the callout. Possible values: BOOL, NUM, TEXT.

httpMethod

Method used in the HTTP request that this callout sends. Mutually exclusive with -fullReqExpr. Possible values: GET, POST.

hostExpr

Text expression to configure the Host header. The expression can contain a literal value (10.101.10.11) or a derived value (for example, HTTP.REQ.HEADER). The literal value can be an IP address or a fully qualified domain name.

Mutually exclusive with -fullReqExpr. Maximum length: 255.

urlStemExpr

A string expression for generating the URL stem. The expression can contain a literal string (for example, /mysite/index.html) or an expression that derives the value (for example, HTTP.REQ.URL).

Mutually exclusive with -fullReqExpr. Maximum length: 8191.

headers

Text expression to insert HTTP headers and their values in the HTTP callout request. You must specify a value for every header. You specify the header name as a string and the header value as an expression. You can configure a maximum of eight headers for an HTTP callout.

Mutually exclusive with -fullReqExpr.

parameters

Expression to insert query parameters in the HTTP request that the callout sends.

You must specify a value for every parameter that you configure. If the callout request uses the GET method, these parameters are inserted in the URL. If the callout request uses the POST method, these parameters are inserted in the POST body.

You configure the query parameter name as a string, and the value as an expression. The parameter values are URL encoded. Mutually exclusive with -fullReqExpr.

fullReqExpr

Exact HTTP request that the NetScaler appliance is to send, as an expression. If you specify this parameter, you must omit the httpMethod, hostExpr, urlStemExpr, headers, and parameters arguments.

The request expression is constrained by the feature for which the callout is used. For example, an HTTP.RES expression cannot be used in a request-time policy bank or in a TCP content switching policy bank.

The NetScaler appliance does not check the validity of this request. You must manually validate the request. The maximum length of the input expression is 8191 characters. The maximum length of a string literal that can be used inside the expression is 255 characters. A string literal that contains more than 255 characters can be split into smaller strings of 255 characters each. The smaller strings can then be concatenated with the + operator to obtain the original string literal. For example, if a string is 500 characters long, you can split the string into smaller strings and concatenate them as shown:

```
'"<string of length 255 characters>" + "<string of length 245
characters>"'
```

resultExpr

Expression that extracts objects from the response sent by the HTTP callout agent. The expression must be a response-based expression, that is, it must begin with `HTTP.RES`.

The operations in this expression must match the return type. For example, if you configure a return type of `TEXT`, the result expression must be a text-based expression. If the return type is `NUM`, the result expression (`resultExpr`) must return a numeric value, as in the following example:

```
HTTP.RES.BODY(10000).LENGTH
```

Maximum length: 8191.

To configure an HTTP callout by using the NetScaler configuration utility

- In the navigation pane of the NetScaler configuration utility, expand **AppExpert**, and then click **HTTP Callouts**.
- In the details pane, do one of the following:
  - To create an HTTP callout, click **Add**.
  - To modify an existing HTTP callout, select the **HTTP callout**, and then click **Open**.
- In the **Create HTTP Callout** dialog box or **Configure HTTP Callout** dialog box, specify values for the following parameters, which correspond to the parameters described in the "Parameters for configuring an HTTP callout" as shown:
  - **Name*** —name (cannot be changed for an existing HTTP callout).
  - **Virtual Server**—vServer
  - **IP Address**—IPAddress
  - **IPv6**: If you want to enter an IPv6 address, select this check box.
  - **Port**—port
- If you want to configure an attribute-based HTTP callout, click **Attribute-based**, click **Configure Request Attributes**, and then, in the **Configure HTTP Callout Request Attributes** dialog box, specify values for the following parameters, which correspond to the parameters described in the "Parameters for configuring an HTTP callout" as shown:
  - **Method**—httpMethod
  - **Host Expression**—hostExpr
  - **URL Stem Expression**—urlStemExpr
  - **Headers**—headers (After you enter a header name and its value, click **Add**, and then enter the next header-value pair.)
  - **Parameters**—parameters (After you enter a parameter name and its value, click **Add**, and then enter the next parameter-value pair.)
- If you want to configure an expression-based HTTP callout, click **Expression-based**, and then, in the space provided, configure the expression. This expression corresponds to the -**fullReqExpr** parameter that is described in "Parameters for configuring an HTTP callout."
- In the **Server Response** area, specify values for the following parameters, which correspond to the parameters described in the "Parameters for configuring an HTTP callout" as shown:
  - **Return Type**—returnType
  - **Expression to extract data from the response**—resultExpr
- Click **Create** or **OK**.

- Verify that the HTTP callout is correctly configured by selecting the callout and viewing the **Details** section at the bottom of the details pane.

Verifying the Configuration

For an HTTP callout to work correctly, all the HTTP callout parameters and the entities associated with the callout must be configured correctly. While the NetScaler appliance does not check the validity of the HTTP callout parameters, it indicates the state of the bound entities, namely the server or virtual server to which the HTTP callout is sent. The following table lists the icons and describes the conditions under which the icons are displayed.

Table 1. Icons That Indicate the States of Entities Bound to an HTTP Callout

Icon	Indicates that
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is UP.
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is OUT OF SERVICE.
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is DOWN.

For an HTTP callout to function correctly, the icon must be green at all times. If the icon is not green, check the state of the callout server or virtual server to which the HTTP callout is sent. If the HTTP callout is not working as expected even though the icon is green, check the parameters configured for the callout.

You can also verify the configuration by sending test requests that match the policy from which the HTTP callout is invoked, checking the hits counter for the policy and the HTTP callout, and verifying the responses that the NetScaler appliance sends to the client.

Note: An HTTP callout can sometimes invoke itself recursively a second time. If this happens, the hits counter is incremented by two counts for each callout that is generated by the appliance. For the hits counter to display the correct value, you must configure the HTTP callout in such a way that it does not invoke itself a second time. For more information about how you can avoid HTTP callout recursion, see Avoiding HTTP Callout Recursion.

To view the hits counter for an HTTP callout

1. In the configuration utility, in the navigation pane, expand **AppExpert**, and then click **HTTP Callouts**.
2. In the details pane, click the **HTTP callout** for which you want to view the hits counter, and then view the hits in the **Details** area.

Invoking an HTTP Callout

After you configure an HTTP callout, you invoke the callout by including the `SYS.HTTP_CALLOUT(<name>)` expression in a default syntax policy rule. In this expression, <name> is the name of the HTTP callout that you want to invoke.

You can use default syntax expression operators with the callout expression to process the response and then perform an appropriate action. The return type of the response from the HTTP callout agent determines the set of operators that you can use on the response. If the part of the response that you want to analyze is text, you can use a text operator to analyze the response. For example, you can use the `CONTAINS(<string>)` operator to check whether the specified portion of the response contains a particular string, as in the following example:

SYS.HTTP_CALLOUT(mycallout).contains("Good IP address")

If you use the preceding expression in a responder policy, you can configure an appropriate responder action.

Similarly, if the part of the response that you want to evaluate is a number, you can use a numeric operator such as GT(int). If the response contains a Boolean value, you can use a Boolean operator.

Note: An HTTP callout can invoke itself recursively. HTTP callout recursion can be avoided by combining the HTTP callout expression with a default syntax expression that prevents recursion. For information about how you can avoid HTTP callout recursion, see Avoiding HTTP Callout Recursion.

You can also cascade HTTP callouts by configuring policies that each invoke a callout after evaluating previously generated callouts. In this scenario, after one policy invokes a callout, when the NetScaler appliance is parsing the callout before sending the callout to the callout server, a second set of policies can evaluate the callout and invoke additional callouts, which can in turn be evaluated by a third set of policies, and so on. Such an implementation is described in the following example.

First, you could configure an HTTP callout called myCallout1, and then configure a responder policy, Pol1, to invoke myCallout1. Then, you could configure a second HTTP callout, myCallout2, and a responder policy, Pol2. You configure Pol2 to evaluate myCallout1 and invoke myCallout2. You bind both responder policies globally.

To avoid HTTP callout recursion, myCallout1 is configured with a unique custom HTTP header called "Request1." Pol1 is configured to avoid HTTP callout recursion by using the default syntax expression,

HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\").NOT.

Pol2 uses the same default syntax expression, but excludes the .NOT operator so that the policy evaluates myCallout1 when the NetScaler appliance is parsing it. Note that myCallout2 identifies its own unique header called "Request2," and Pol2 includes a default syntax expression to prevent myCallout2 from invoking itself recursively.

Example

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -returnType TEXT -hostExpr
 "\"10.102.3.95\"" -urlStemExpr "\"/cgi-bin/check_clnt_from_database.pl\"" -headers Request1
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"

Done

> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\").NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS(\"IP Matched\")" RESET

Done

> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

> set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -returnType TEXT -hostExpr
"\"10.102.3.96\"" -urlStemExpr "\"/cgi-bin/check_clnt_location_from_database.pl\"" -headers Request2
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(200)"

 Done

> add responder policy Pol2 "HTTP.REQ.HEADER(\"Request2\").EQ(\"Callout Request\").NOT &&
 HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\") && SYS.HTTP_CALLOUT(myCallout2).CONTAINS
(\"APAC\")" RESET

 Done

> bind responder global Pol2 110 END -type OVERRIDE

Done
```

Avoiding HTTP Callout Recursion

Even though the NetScaler appliance does not check for the validity of the HTTP callout request, it parses the request once before it sends the request to the HTTP callout agent. This parsing enables the NetScaler appliance to treat the callout request as it does any other incoming request, which in turn enables you to configure several useful NetScaler features (such as the integrated cache, SureConnect, and Priority Queuing) to work on the callout request.

However, during this parsing, the HTTP callout request can hit the same policy and therefore invoke itself recursively. The NetScaler appliance detects the recursive invocation and raises an undefined (UNDEF) condition. However, the recursive invocation results in the policy and HTTP callout hit counters being incremented by two counts each instead of one count each.

To prevent a callout from invoking itself, you must identify at least one unique characteristic of the HTTP callout request, and then exclude all requests with this characteristic from being processed by the policy rule that invokes the callout. You can do so by including another default syntax expression in the policy rule. The expression must precede the `SYS.HTTP_CALLOUT(<name>)` expression so that it is evaluated before the callout expression is evaluated. For example:

<Expression that prevents callout recursion> && SYS.HTTP_CALLOUT(<name>)

When you configure a policy rule in this way, when the NetScaler appliance generates the request and parses it, the compound rule evaluates to FALSE, the callout is not generated a second time, and the hit counters are incremented correctly.

One way by which you can assign a unique characteristic to an HTTP callout request is to include a unique custom HTTP header when you configure the callout. Following is an example of an HTTP callout called "myCallout." The callout generates an HTTP request that checks whether a client's IP address is present in a database of blacklisted IP addresses. The callout includes a custom header called "Request," which is set to the value "Callout Request." A globally bound responder policy, "Pol1," invokes the HTTP callout but excludes all requests whose Request header is set to this value, thus preventing a second invocation of myCallout. The expression that prevents a second invocation is `HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT`.

Example

```
> add policy httpCallout myCallout
 Done
> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -returnType TEXT -hostExpr "\"10.102.3.95
Done
> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALL
Done
> bind responder global Pol1 100 END -type OVERRIDE
Done
```

Note: You can also configure an expression to check whether the URL of the request includes the URL stem expression that is configured for the HTTP callout. If you want to implement this scenario, make sure that the HTTP callout agent is dedicated to respond only to HTTP callouts and not to other client requests directed through the NetScaler appliance. If the HTTP callout agent is an application or Web server that serves other client requests, such an expression will prevent the NetScaler appliance from processing those client requests. Instead, use a unique custom header as described earlier.

Deployment Scenarios for HTTP Callouts

These topics demonstrate the configuration of HTTP callouts to perform various useful tasks. In all cases, the NetScaler appliance performs a callout to an external server where a callout agent is configured to respond to the request from the NetScaler appliance on the basis of data that is present on the external server. The callout agent is a program, for example, a CGI script, that is deployed in a container (for example, Apache Tomcat).

Note that the examples should be used only as a guideline when you want to create scripts that work in your deployment. The IP addresses and other entities used in these deployment scenarios should be modified to suit your environment.

Example 1: Filtering Clients by Using an IP Blacklist

HTTP callouts can be used to block requests from clients that are blacklisted by the administrator. The list of clients can be a publicly known blacklist, a blacklist that you maintain for you organization, or a combination of both.

The NetScaler appliance checks the IP address of the client against the pre-configured blacklist and blocks the transaction if the IP address has been blacklisted. If the IP address is not in the list, the appliance processes the transaction.

To implement this configuration, you must perform the following tasks:

1. Enable responder on the NetScaler appliance.
2. Create an HTTP callout on the NetScaler appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response to the HTTP callout, and then bind the policy globally.
4. Create an HTTP callout agent on the remote server.

Enabling Responder

You must enable responder before you can use it.

To enable responder by using the configuration utility

1. Make sure that you have installed the responder license.
2. In the navigation pane, right-click **Responder**, and then click **Enable Responder feature**.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTP-Callout-1, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

Table 1. Parameters and Values for HTTP-Callout-1

Parameter	Value
Name	HTTP-Callout-1
Server to receive callout request	
IP Address	10.103.9.95
Port	80
Request to send to the server	
Method	GET
Host Expression	10.102.3.95
URL Stem Expression	"/cgi-bin/check_clnt_from_database.pl"
Headers	
Name	Request
Value-expression	Callout Request
Parameters	
Name	Cip
Value-expression	CLIENT.IP.SRC
Server Response	
Return Type	TEXT
Expression to extract data from the response	HTTP.RES.BODY(100)

Configuring a Responder Policy and Binding it Globally

After you configure the HTTP callout, verify the callout configuration, and then configure a responder policy to invoke the callout. While you can create a responder policy in the **Policies** sub-node and then bind it globally by using the **Responder Policy Manager**, this demonstration uses the **Responder Policy Manager** to create the responder policy and bind the policy globally.

To create a responder policy and bind it globally by using the configuration utility

1. In the navigation pane, click **Responder**.
2. In the details pane, under **Policy Manager**, click **Policy Manager**.
3. In the **Responder Policy Manager** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, under **Policy Name**, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
    a. In **Name**, type **Policy-Responder-1**.
    b. In **Action**, select **RESET**.
    c. In **Undefined-Result Action**, select **Global undefined-result action**.
    d. In **Expression**, type the following default syntax expression:

        "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Callout-1

    e. Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout Agent on the Remote Server

You must now create an HTTP callout agent on the remote callout server that will receive callout requests from the NetScaler appliance and respond appropriately. The HTTP callout agent is a script that is different for each deployment and must be written with the server specifications in mind, such as the type of database and the scripting language supported.

Following is a sample callout agent that verifies whether the given IP address is part of an IP blacklist. The agent has been written in the Perl scripting language and uses a MYSQL database.

The following CGI script checks for a given IP address on the callout server.

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
	use DBI();
	use CGI qw(:standard);
#Take the Client IP address from the request query
	my $ip_to_check = param('cip');
# Where a MYSQL database is running
	my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
# Database username to connect with
	my $db_user_name = ‘dbuser’;
# Database password to connect with
	my $db_password = 'dbpassword';
	my ($id, $password);
# Connecting to the database
	my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
	my $sth = $dbh->prepare(qq{ select * from bad_clnt });
	$sth->execute();
	while (my ($ip_in_database) = $sth->fetchrow_array()) {
	chomp($ip_in_database);
# Check for IP match
	if ($ip_in_database eq $ip_to_check) {
	    print "\n IP Matched\n";
                                        $sth->finish();
                                    exit;
	     }
	}
	print "\n IP Failed\n";
	$sth->finish();
	exit;
```

Example 2: ESI Support for Fetching and Updating Content Dynamically

Edge Side Includes (ESI) is a markup language for edge-level dynamic Web content assembly. It helps in accelerating dynamic Web-based applications by defining a simple markup language to describe cacheable and non-cacheable Web page components that can be aggregated, assembled, and delivered at the network edge. By using HTTP callouts on the NetScaler appliance, you can read through the ESI constructs and aggregate or assemble content dynamically.

To implement this configuration, you must perform the following tasks:

1. Enable rewrite on the NetScaler appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a rewrite action to replace the ESI content with the callout response body.
4. Configure a rewrite policy to specify the conditions under which the action is performed, and then bind the rewrite policy globally.

Enabling Rewrite

Rewrite must be enabled before it is used on the NetScaler appliance. The following procedure describes the steps to enable the rewrite feature.

To enable rewrite by using the configuration utility

1. Make sure that you have installed the rewrite license.
2. In the navigation pane, right-click **Rewrite**, and then click **Enable Rewrite feature**.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTP-Callout-2, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

Table 1. Parameters and Values for HTTP-Callout-2

Parameter	Value
Name	HTTP-Callout-2
Server to receive callout request	
IP Address	10.102.56.51
Port	80
Request to send to the server	
Method	GET
Host Expression	10.102.56.51:80
URL Stem Expression	"HTTP.RES.BODY(500).AFTER_STR(\"src=\").BEFORE_STR(\"/>\")"
Headers	
Name	Name
Value-expression	Callout
Server Response	
Return Type	TEXT
Expression to extract data from the response	HTTP.RES.BODY(100)

Configuring the Rewrite Action

Create a rewrite action, Action-Rewrite-1, to replace the ESI content with the callout response body. Use the parameter settings shown in the following table.

Table 1. Parameters and Values for Action-Rewrite-1

Parameter	Value
Name	Action-Rewrite-1
Type	Replace
Expression to choose target text reference	"HTTP.RES.BODY(500).AFTER_STR (\" <example>\").BEFORE_STR (\"</example>\")"
String expression for replacement text	"SYS.HTTP_CALLOUT(HTTP-Callout-2)"

To configure the rewrite action by using the configuration utility

1. In the navigation pane, expand **Rewrite**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Rewrite Action** dialog box, in **Name**, type **Action-Rewrite-1**.
4. In **Type**, select **REPLACE**.
5. In **Expression to choose target text reference**, type the following default syntax expression:

   "HTTP.RES.BODY(500).AFTER_STR(\"<example>\").BEFORE_STR(\"</example>\")"
6. In the **String expression for replacement text**, type the following string expression:

   "SYS.HTTP_CALLOUT(HTTP-Callout-2)"
7. Click **Create**, and then click **Close**.

Creating the Rewrite Policy and Binding it Globally

Create a rewrite policy, Policy-Rewrite-1, with the parameter settings shown in the following table. You can create a rewrite policy in the **Policies** subnode and then bind it globally by using the **Rewrite Policy Manager**. Alternatively, you can use the **Rewrite Policy Manager** to perform both these tasks simultaneously. This demonstration uses the **Rewrite Policy Manager** to perform both tasks.

Table 1. Parameters and Values for Policy-Rewrite-1

Parameter	Value
Name	Policy-Rewrite-1
Action	Action_Rewrite-1
Undefined Result Action	-Global undefined-result action-
Expression	"HTTP.REQ.HEADER(\"Name\").CONTAINS (\"Callout\").NOT"

To configure a rewrite policy and bind it globally by using the configuration utility

1. In the navigation pane, expand **Rewrite**.
2. In the details pane, under **Policy Manager**, click **Rewrite Policy Manager**.
3. In the **Rewrite Policy Manager** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Rewrite Policy** dialog box, do the following:
    a. In **Name**, type **Policy-Rewrite-1**.
    b. In **Action**, select **Action-Rewrite-1**.
    c. In **Undefined-Result Action**, select **Global undefined-result action**.
    d. In **Expression**, type the following default syntax expression:

        "HTTP.REQ.HEADER(\"Name\").CONTAINS(\"Callout\").NOT"

    e. Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Example 3: Access Control and Authentication

In high security zones, it is mandatory to externally authenticate the user before a resource is accessed by clients. On the NetScaler appliance, you can use HTTP callouts to externally authenticate the user by evaluating the credentials supplied. In this example, the assumption is that the client is sending the user name and password through HTTP headers in the request. However, the same information could be fetched from the URL or the HTTP body.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the NetScaler appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling Responder

The responder feature must be enabled before it is used on the NetScaler appliance.

To enable responder by using the configuration utility

1. Make sure that the responder license is installed.
2. In the navigation pane, right-click **Responder**, and then click **Enable Responder feature**.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTP-Callout-3, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

Table 1. Parameters and Values for HTTP-Callout-3

Parameter	Value
Name	HTTP-Callout-3
Server to receive callout request	
IP Address	10.103.9.95
Port	80
Request to send to the server	
Method	GET
Host Expression	10.102.3.95
URL Stem Expression	“/cgi-bin/authenticate.pl”
Headers	
Name	Request
Value-expression	Callout Request
Parameters	
Name	Username
Value-expression	HTTP.REQ.HEADER("Username").VALUE(0)
Name	Password
Value-expression	HTTP.REQ.HEADER("Password").VALUE(0)
Server Response	
Return Type	TEXT
Expression to extract data from the response	HTTP.RES.BODY(100)

Creating a Responder Policy to Analyze the Response

Create a responder policy, Policy-Responder-3, that will check the response from the callout server and RESET the connection if the source IP address has been blacklisted. Create the policy with the parameters settings shown in the following table. While you can create a responder policy in the **Policies** subnode and then bind it globally by using the **Responder Policy Manager**, this demonstration uses the **Responder Policy Manager** to create the responder policy and bind the policy globally.

Table 1. Parameters and Values for Policy-Responder-3

Parameter	Value
Name	Policy-Responder-3
Action	RESET
Undefined-Result-Action	-Global undefined-result action-
Expression	"HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\"Authentication Failed\")"

To create a responder policy and bind it globally by using the configuration utility

1. In the navigation pane, expand **Responder**.
2. In the details pane, under **Policy Manager**, click **Responder Policy Manager**.
3. In the **Responder Policy Manger** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
    a. In **Name**, type **Policy-Responder-3**.
    b. In **Action**, select **RESET**.
    c. In **Undefined-Result Action** , select **Global undefined-result action**.
    d. In the **Expression** text box, type:

    "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Call

    e. Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout Agent on the Remote Server

You now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the NetScaler appliance and responds appropriately. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

Following is sample callout agent pseudo-code that verifies whether the supplied user name and password are valid. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To verify the supplied user name and password by using pseudo-code

1. Accept the user name and password supplied in the request and format them appropriately.
2. Connect to the database that contains all the valid user names and passwords.
3. Check the supplied credentials against your database.
4. Format the response as required by the HTTP callout.
5. Send the response to the NetScaler appliance.

Example 4: OWA-Based Spam Filtering

Spam filtering is the ability to dynamically block emails that are not from a known or trusted source or that have inappropriate content. Spam filtering requires an associated business logic that indicates that a particular kind of message is spam. When the NetScaler appliance processes Outlook Web Access (OWA) messages based on the HTTP protocol, HTTP callouts can be used to filter spam.

You can use HTTP callouts to extract any portion of the incoming message and check with an external callout server that has been configured with rules that are meant for determining whether a message is legitimate or spam. In case of spam email, for security reasons, the NetScaler appliance does not notify the sender that the email is marked as spam.

The following example conducts a very basic check for various listed keywords in the email subject. These checks can be more complex in a production environment.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the NetScaler appliance.
2. Create an HTTP callout on the NetScaler appliance and configure it with details about the external server and other required parameters.
3. Create a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling Responder

The responder feature must be enabled before it can be used on the NetScaler appliance.

To enable responder by using the configuration utility

1. Make sure that the responder license is installed.
2. In the navigation pane, right-click **Responder**, and then click **Enable Responder feature**.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTP-Callout-4, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

Table 1. Parameters and Values for HTTP-Callout-4

Parameter	Value
Name	HTTP-Callout-4
Server to receive callout request	
IP Address	10.103.56.51
Port	80
Request to send to the server	
Method	POST
Host Expression	fffffff
URL Stem Expression	"/cgi-bin/Callout/spam_filter.pl"
Headers	
Name	Request
Value-expression	Callout Request
Parameters	
Name	Subject
Value-expression	("\"" + HTTP.REQ.BODY(1000).AFTER_STR("urn:schemas:httpmail:subject=").BEFORE_STR("\n").TO_LOWER + "\"")
Server Response	
Return Type	BOOL
Expression to extract data from the response	HTTP.RES.BODY(100) .CONTAINS(\"Matched\")

Creating a Responder Action

Create a responder action, Action-Responder-4. Create the action with the parameter settings shown in the following table.

Table 1. Parameters and Values for Action-Responder-4

Parameter	Value
Name	Action-Responder-4
Type	Respond with
Target	"\"HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n\""

To create a responder action by using the configuration utility

1. In the navigation pane, expand **Responder**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, in **Name**, type **Action-Responder-4**.
4. In **Type**, click **Respond with**.
5. In **Target**, type:

   "\"HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-W
6. Click **Create**, and then click **Close**.

Creating a Responder Policy to Invoke the HTTP Callout

Create a responder policy, Policy-Responder-4, that will check the request body and, if the body contains the word "*subject*," invoke the HTTP callout to verify the email. Create the policy with the parameter settings shown in the following table. While you can create a responder policy in the **Policies** subnode and then bind it globally by using the **Responder Policy Manager**, this demonstration uses the **Responder Policy Manager** to create the responder policy and bind it globally.

Table 1. Parameters and Values for Policy-Responder-4

Parameter	Value
Name	Policy-Responder-4
Action	Action-Responder-4
Undefined-Result-Action	-Global undefined-result action-
Expression	"HTTP.REQ.BODY(1000).CONTAINS(\"urn:schemas:httpmail:subject\") && SYS.HTTP_CALLOUT(HTTP-Callout-4)"

To create a responder policy by using the configuration utility

1. In the navigation pane, click **Responder**.
2. In the details pane, under **Policy Manager**, click **Responder policy manager**.
3. In the **Responder Policy Manger** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
   a. In **Name**, type **Policy-Responder-4**.
   b. In **Action**, click **Action-Responder-4**.
   c. In **Undefined-Result Action**, click **Global undefined-result action**.
   d. In the **Expression** text box, type:

      "HTTP.REQ.BODY(1000).CONTAINS(\"urn:schemas:httpmail:subject\") && SYS.HTTP_CALLOUT(HTTP-C

   e. Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout Agent on the Remote Server

You will now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the NetScaler appliance and responds accordingly. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

The following pseudo-code provides instructions for creating a callout agent that checks a list of words that are generally understood to indicate spam mails. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To identify spam email by using pseudo-code

1. Accept the email subject provided by the NetScaler appliance.
2. Connect to the database that contains all the terms against which the email subject is checked.
3. Check the words in the email subject against the spam word list.
4. Format the response as required by the HTTP callout.
5. Send the response to the NetScaler appliance.

HTTP Denial-of-Service Protection

Internet hackers can bring down a site by sending a surge of GET requests or other HTTP-level requests. HTTP Denial-of-Service (HTTP Dos) Protection provides an effective way to prevent such attacks from being relayed to your protected Web servers. The HTTP DoS feature also ensures that a NetScaler appliance located between the internet cloud and your Web servers is not brought down by an HTTP DoS attack.

Most attackers on the Internet use applications that discard responses to reduce computation costs, and minimize their size to avoid detection. The attackers focus on speed, devising ways to send attack packets, establish connections or send HTTP requests as rapidly as possible.

Real HTTP clients such as Internet Explorer, Firefox, or NetScape browsers can understand HTML Refresh meta tags, Java scripts, and cookies. In standard HTTP the clients have most of these features enabled. However, the dummy clients used in DoS attacks cannot parse the response from the server. If malicious clients attempt to parse and send requests intelligently, it becomes difficult for them to launch the attack aggressively.

When the NetScaler appliance detects an attack, it responds to a percentage of incoming requests with a Java or HTML script containing a simple refresh and cookie. (You configure that percentage by setting the Client Detect Rate parameter.) Real Web browsers and other Web-based client programs can parse this response and then resend a POST request with the cookie. DoS clients drop the NetScaler appliance's response instead of parsing it, and their requests are therefore dropped as well.

Even when a legitimate client responds correctly to the NetScaler appliance's refresh response, the cookie in the client's POST request may become invalid in the following conditions:

- If the original request was made before the NetScaler appliance detected the DoS attack, but the resent request was made after the appliance had come under attack.
- When the client's think time exceeds four minutes, after which the cookie becomes invalid.

Both of these scenarios are rare, but not impossible. In addition, the HTTP DoS protection feature has the following limitations:

- Under an attack, all POST requests are dropped, and an error page with a cookie is sent.
- Under an attack, all embedded objects without a cookie are dropped, and an error page with a cookie is sent.

The HTTP DoS protection feature may affect other NetScaler features. Using DoS protection for a particular content switching policy, however, creates additional overhead because the policy engine must find the policy to be matched. There is some overhead for SSL requests due to SSL decryption of the encrypted data. Because most attacks are not on a secure network, though, the attack is less aggressive.

If you have implemented priority queuing, while it is under attack a NetScaler appliance places requests without proper cookies in a low-priority queue. Although this creates overhead, it protects your Web servers from false clients. HTTP DoS protection typically has minimal effect on throughput, since the test JavaScript is sent for a small percentage of requests only. The latency of requests is increased, because the client must re-issue the request after it receives the JavaScript. These requests are also queued

To implement HTTP DoS protection, you enable the feature and define a policy for applying this feature. Then you configure your services with the settings required for HTTP DoS. You also bind a TCP monitor to each service and bind your policy to each service to put it into effect.

Layer 3-4 SYN Denial-of-Service Protection

Any NetScaler appliance with system software version 8.1 or later automatically provides protection against SYN DoS attacks.

To mount such an attack, a hacker initiates a large number of TCP connections but does not respond to the SYN-ACK messages sent by the victimized server. The source IP addresses in the SYN messages received by the server are typically spoofed. Because new SYN messages arrive before the half-open connections initiated by previous SYN messages time out, the number of such connections increases until the server no longer has enough memory available to accept new connections. In extreme cases, the system memory stack can overflow.

A NetScaler appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted.

SYN DoS protection on the NetScaler appliance ensures the following:

- The memory of the NetScaler is not wasted on false SYN packets. Instead, memory is used to serve legitimate clients.
- Normal TCP communications with legitimate clients continue uninterrupted, even when the Web site is under SYN flood attack.

In addition, because the NetScaler appliance allocates memory for HTTP connection state only after it receives an HTTP request, it protects Web sites from idle connection attacks.

SYN DoS protection on your NetScaler appliance requires no external configuration. It is enabled by default.

Layer 3-4 SYN Denial-of-Service Protection

Any NetScaler appliance with system software version 8.1 or later automatically provides protection against SYN DoS attacks.

To mount such an attack, a hacker initiates a large number of TCP connections but does not respond to the SYN-ACK messages sent by the victimized server. The source IP addresses in the SYN messages received by the server are typically spoofed. Because new SYN messages arrive before the half-open connections initiated by previous SYN messages time out, the number of such connections increases until the server no longer has enough memory available to accept new connections. In extreme cases, the system memory stack can overflow.

A NetScaler appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted.

SYN DoS protection on the NetScaler appliance ensures the following:

- The memory of the NetScaler is not wasted on false SYN packets. Instead, memory is used to serve legitimate clients.
- Normal TCP communications with legitimate clients continue uninterrupted, even when the Web site is under SYN flood attack.

In addition, because the NetScaler appliance allocates memory for HTTP connection state only after it receives an HTTP request, it protects Web sites from idle connection attacks.

SYN DoS protection on your NetScaler appliance requires no external configuration. It is enabled by default.

Enabling HTTP DoS Protection

To configure HTTP DoS protection, you must first enable the feature.

To enable HTTP DoS protection by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable HTTP DoS protection and verify the configuration:

- enable ns feature HttpDoSProtection
- show ns feature

Example

```
> enable ns feature HttpDoSProtection
 Done
> show ns feature

      Feature                  Acronym           Status
      -------                 -------          ------
 1)   Web Logging               WL               ON
 2)   Surge Protection          SP               OFF
 .
 .
 .
 10)  Global Server Load Balancing  GSLB            ON
 11)  Http DoS Protection        HDOSP            ON
 12)  Content Filtering          CF               ON
 .
 .
 23)  HTML Injection             HTMLInjection    ON
 24)  NetScaler Push             push             OFF
 Done
>
```

To enable HTTP DoS protection by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, click **Change Advanced Features**.
3. In the **Configure Advanced Features** dialog box, select **HTTP DoS Protection** check box.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the feature has been enabled.

Defining an HTTP DoS Policy

After you enable HTTP DoS protection, you next create a policy.

Note: Before changing the default setting for cltDetectRate, see Tuning the Client Detection/JavaScript Challenge Response Rate.

To configure a HTTP DoS policy by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to configure an HTTP DoS policy and verify the configuration:

- add dos policy <name> **-qDepth** <positive_integer> [**-cltDetectRate** <positive_integer>]
- set dos policy <name> **-qDepth** <positive_integer> [**-cltDetectRate** <positive_integer>]

Example

```
> add dos policy pol-HTTP-DoS  -qDepth 30
 Done
> set dos policy pol-HTTP-DoS  -qDepth 40
 Done
> show dos policy
1)      Policy: pol-HTTP-DoS    QDepth: 40
 Done
>
```

Parameters for defining an HTTP DoS policy

name

A name for your HTTP DoS policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the type of action.

qdepth

An integer that represents the maximum number of connections that can be placed in the queue at one time.

cltDetectRate

An integer that represents the percentage of traffic to which the HTTP DoS policy should be applied.

To configure an HTTP DoS policy by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then click **HTTP DoS**.
2. In the details pane, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Open**.
3. In the **Create HTTP DoS Policy** or **Configure HTTP DoS Policy** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for defining an HTTP DoS policy" as follows (asterisk indicates a required parameter):
   - Name*—name (You cannot change the name of an existing policy.)
   - QDepth*—qdepth
   - Client Detect Rate—cltDetectRate (Before changing the default setting for cltDetectRate, see Tuning the Client Detection/JavaScript Challenge Response Rate.)
4. Click **OK** to create your new policy. The policy that you created appears in the details pane, and the status bar displays a message indicating that the DoS policy is successfully configured.

Configuring an HTTP DoS Service

After you configure an HTTP DoS policy, you must configure a service for your policy. The service accepts HTTP traffic that is protected by the HTTP DoS policy.

To configure an HTTP DoS service by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to configure an HTTP DoS service and verify the configuration:

- add service <name>@ (<IP>@ | <serverName>@) HTTP <port> [**-maxClient** <positive_integer>] [**-maxReq** <positive_integer>] **-state** ENABLED
- set service <name>@ (<IP>@ | <serverName>@) HTTP <port> [**-maxClient** <positive_integer>] [**-maxReq** <positive_integer>] **-state** ENABLED

Example

```
> add service ser-HTTP-Dos1 10.102.29.40 HTTP 87
 Done
> set service ser-HTTP-Dos1 -maxReq 20
 Done
> show service
1)      srv-http-10 (10.102.29.30:80) - HTTP
        State: DOWN
        Last state change was at Wed Jul  8 07:49:52 2009
        Time since last state change: 34 days, 00:48:18.700
        Server Name: 10.102.29.30
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
         .
         .
         .
```

```
5)     ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
       State: DOWN
       Last state change was at Tue Aug 11 08:23:40 2009
       Time since last state change: 0 days, 00:14:30.300
       Server Name: 10.102.29.40
       Server ID : 0   Monitor Threshold : 0
       Max Conn: 0     Max Req: 20     Max Bandwidth: 0 kbits
       Use Source IP: NO
       Client Keepalive(CKA): NO
       Access Down Service: NO
       TCP Buffering(TCPB): NO
       HTTP Compression(CMP): YES
       Idle timeout: Client: 180 sec   Server: 360 sec
       Client IP: DISABLED
       Cacheable: NO
       SC: OFF
       SP: OFF
       Down state flush: ENABLED
 Done
>
```

Parameters for configuring an HTTP DoS service

name

A name for your service. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the traffic this service will handle.

IP

The IP of the server that the service represents.

serverName

The FQDN of the server that the service represents.

port

The port on which your service will listen. This is normally port 80 (for HTTP) or port 443 (for HTTPS).

maxClient

The maximum number of clients.

maxReq

The maximum number of requests that can be sent on a persistent connection to the service.

state

The state of the service after it is added.

To configure an HTTP DoS service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, do one of the following:
   - To create a new service, click **Add**.
   - To modify an existing service, select the service, and then click **Open**.
3. In the **Create Server** or **Configure Server** dialog box, specify values for the following parameters, which correcpond to the descriptions in "Parameters for configuring an HTTP DoS service" as follows (asterisk indicates a required parameter):
   - Service Name*—name (You cannot change the name of an existing service.)
   - Server*—IP or serverName (Specify one or the other, not both.)
   - Port*—port
4. If the **Enable Service** check box is not selected, select it.
5. Select the **Advanced** tab, and select the **Override Global** check box to enable those choices.
6. Specify values for the following parameters.
   - Max Clients*—maxClient
   - Max Requests*—maxReq
7. Click **Create** or **OK**, and then click **Close**. The service appears in the list of services.

Binding an HTTP DoS Monitor and Policy

To put HTTP DoS protection into effect after you have configured an HTTP DoS service, you must bind the monitor, and then bind the service to the HTTP DoS policy.

To bind the monitor to the service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind the monitor to the service and verify the configuration:

- bind lb monitor <monitorName> <serviceName>
- show lb monitor

Example

```
> bind lb monitor tcp ser-HTTP-DoS
 Done
> show lb monitor
1)  Name.......:  ping-default  Type......:    PING   State....ENABLED
2)  Name.......:   tcp-default  Type......:     TCP   State....ENABLED
3)  Name.......:         ping  Type......:    PING   State....ENABLED
4)  Name.......:          tcp  Type......:     TCP   State....ENABLED
5)  Name.......:         http  Type......:    HTTP   State....ENABLED
 .
 .
 .
17)  Name.......:     ldns-dns  Type......: LDNS-DNS   State....ENABLED
 Done
```

To bind the policy to the service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind the policy to the service and verify the configuration:

bind service <serviceName> -policyName <policyname>

Example

```
> bind service ser-HTTP-DoS -policyName pol-HTTP-DoS
 Done
> show service
1)    srv-http-10 (10.102.29.30:80) - HTTP
      State: DOWN
      Last state change was at Wed Jul  8 07:49:52 2009
      Time since last state change: 34 days, 01:24:58.510
      Server Name: 10.102.29.30
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): NO
      Idle timeout: Client: 180 sec   Server: 360 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: ON
      Down state flush: ENABLED
      .
      .
      .
4)    ser-HTTP-Dos (10.102.29.18:88) - HTTP
      State: DOWN
      Last state change was at Tue Aug 11 08:19:45 2009
      Time since last state change: 0 days, 00:55:05.40
      Server Name: 10.102.29.18
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): YES
      Idle timeout: Client: 180 sec   Server: 360 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: ON
      Down state flush: ENABLED
5)    ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
      State: DOWN
      Last state change was at Tue Aug 11 08:23:40 2009
      Time since last state change: 0 days, 00:51:10.110
      Server Name: 10.102.29.40
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 20     Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): YES
      Idle timeout: Client: 180 sec   Server: 360 sec
```

```
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
Done
>
```

To bind the monitor and policy to the service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service that you want to bind, and then click **Open**.
3. In the **Configure Service** dialog box, select the **Monitor** tab, click the name of the monitor you want in the Monitors list, and then click **Add**. The selected monitor is added to the **Configured** frame.
4. Select the **Policies** tab, select a policy from the **Available Policies** list, and then click **Add**. The policy appears in the **Configured Policies** list.
5. Click **OK**, and then click **Close**. A message appears in the status bar, stating that the service has been configured.

Tuning the Client Detection/JavaScript Challenge Response Rate

After you have enabled and configured HTTP DoS protection, if more than the maximum specified number of clients are waiting in the NetScaler surge queue for the HTTP DoS service, the HTTP DoS protection function is triggered. The default rate of challenged JavaScript responses sent to the client is one percent of the server response rate. The default response rate is inadequate in many real attack scenarios, however, and may need to be tuned.

For example, assume that the Web server is capable of a maximum of 500 responses/sec, but is receiving 10,000 Gets/sec. If 1% of the server responses are sent as JavaScript challenges, responses are reduced to almost none: 5 client (500 *0.01) Javascript responses, for 10000 waiting client requests. Only about 0.05% of the real clients receive JavaScript challenge responses. However, if the client detection/Javascript challenge response rate is very high (for example, 10%, generating 1000 challenge Javascript responses per second), it may saturate the upstream links or harm the upstream network devices. Exercise care when modifying the default **Client Detect Rate** value.

If the configured triggering surge queue depth is, for example, 200, and the surge queue size is toggling between 199 and 200, the NetScaler toggles between the “attack” and “no-attack” modes, which is not desirable. The HTTP DoS feature includes a window mechanism is provided. When the surge queue size reaches the designated queue depth value, triggering “attack” mode, the surge queue size must fall for the NetScaler to enter “no-attack” mode. In the scenario just described, if the value of WINDOW_SIZE is set to 20, the surge queue size must fall below 180 before the NetScaler enters “no-attack” mode. During configuration, you must specify a value more than the WINDOW_SIZE for the **QDepth** parameter when adding a DoS policy or setting a DoS policy.

The triggering surge queue depth should be configured on the basis of previous observations of traffic characteristics. For more information about setting up a correct configuration, see Guidelines for HTTP DoS Protection Deployment.

Guidelines for HTTP DoS Protection Deployment

Citrix recommends you to deploy the HTTP DoS protection feature in a tested and planned manner and closely monitor its performance after the initial deployment. Use the following information to fine-tune the deployment of HTTP DoS Protection.

- The maximum number of concurrent connections supported by your servers.
- The average and normal values of the concurrent connections supported by your servers.
- The maximum output rate (responses/sec) that your server can generate.
- The average traffic that your server handles.
- The typical bandwidth of your network.
- The maximum bandwidth available upstream.
- The limits affecting bandwidth (such as external links, a particular router, or other critical devices on the path that may suffer from a traffic surge).
- Whether allowing a greater number of clients to connect is more important than protecting upstream network devices.

To determine the characteristics of a HTTP DoS attack, you should consider the following issues.

- What is the rate of incoming fake requests that you have experienced in the past?
- What types of requests have you received (complete posts, incomplete gets)?
- Did previous attacks saturate your downstream links? If not, what was the bandwidth?
- What types of source IP addresses and source ports did the HTTP requests have (e.g., IP addresses from one subnet, constant IP, ports increasing by one).
- What types of attacks do you expect in future? What type have you seen in the past?
- Any or all information that can help you tune DoS attack protection.

Domain Name System

You can configure the Citrix® NetScaler® appliance to function as an authoritative domain name server (ADNS server) for a domain. You can add the DNS resource records that belong to the domain for which the appliance is authoritative and configure resource record parameters. You can also configure the NetScaler appliance as a proxy DNS server that load balances a farm of DNS name servers that are either within your network or outside your network. You can configure the appliance as an end resolver and forwarder. You can configure DNS suffixes that enable name resolution when fully qualified domain names are not configured. The appliance also supports the DNS ANY query that retrieves all the records that belong to a domain.

You can configure the NetScaler appliance to concurrently function as an authoritative DNS server for one domain and a DNS proxy server for another domain. When you configure the NetScaler as the authoritative DNS server or DNS proxy server for a zone, you can enable the appliance to use the Transmission Control Protocol (TCP) for response sizes that exceed the size limit specified for the User Datagram Protocol (UDP).

How DNS Works on the NetScaler

You can configure the NetScaler appliance to function as an ADNS server, DNS proxy server, end resolver, and forwarder. You can add DNS resource records on the NetScaler, including service (SRV) records, IPv6 (AAAA) records, address (A) records, mail exchange (MX) records, canonical name (CNAME) records, pointer (PTR) records, and start of authority (SOA) records. Also, you can configure the NetScaler to load balance external DNS name servers.

The NetScaler can be configured as the authority for a domain. To do this, you add valid SOA and NS records for the domain.

An ADNS server is a DNS server that contains complete information about a zone.

To configure the NetScaler as an ADNS server for a zone, you must add an ADNS service, and then configure the zone. To do so, you add valid SOA and NS records for the domain. When a client sends a DNS request, the NetScaler appliance searches the configured resource records for the domain name. You can configure the ADNS service to be used with the NetScaler Global Server Load Balancing (GSLB) feature.

You can delegate a subdomain, by adding NS records for the subdomain to the zone of the parent domain. You can then make the NetScaler authoritative for the subdomain, by adding a "glue record" for each of the subdomain name servers. If GSLB is configured, the NetScaler makes a GSLB load balancing decision based on its configuration and replies with the IP address of the selected virtual server. The following figure shows the entities in an ADNS GSLB setup and a DNS proxy setup.

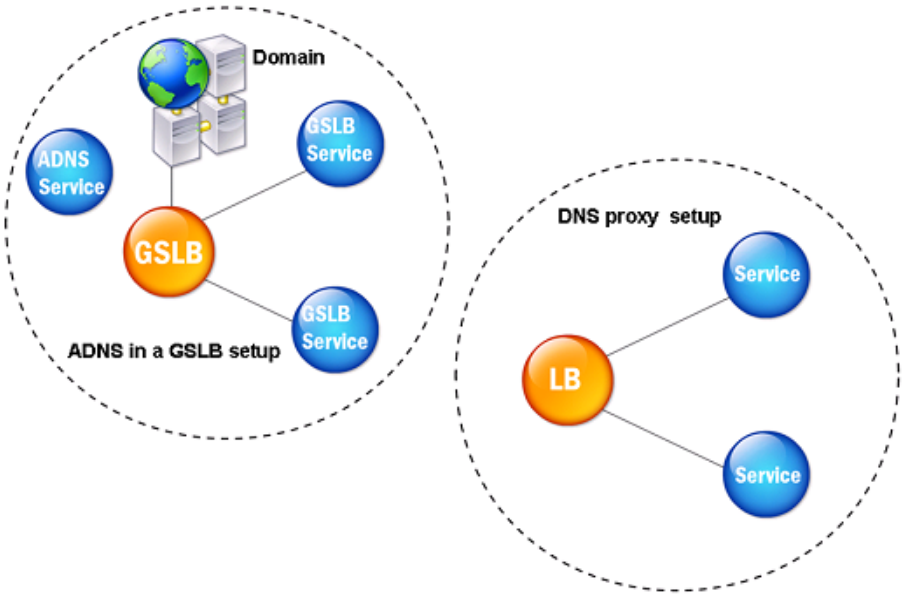


Figure 1. DNS Proxy Entity Model

The NetScaler can cache DNS responses (records) and can function as a DNS proxy. This enables the NetScaler to provide quick responses for repeated translations.

To configure the NetScaler as a DNS proxy, you must enable caching of DNS records. You must also create a load balancing DNS virtual server, and DNS services, and then bind these services to the virtual server.

The NetScaler provides two options, minimum time to live (TTL) and maximum TTL for configuring the lifetime of the cached data. The cached data times out as specified by your settings for these two options. The NetScaler checks the TTL of the DNS record coming from the server. If the TTL is less than the configured minimum TTL, it is replaced with the configured minimum TTL. If the TTL is greater than the configured maximum TTL, it is replaced with the configured maximum TTL.

The NetScaler also allows caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called *negative caching*. Negative caching helps speed up responses to queries on a domain, and can optionally provide the record type.

A negative response can be one of the following:

- NXDOMAIN error message - If a negative response is present in the local cache, the NetScaler returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the NetScaler. The NetScaler caches the response locally, then returns the error message to the client.

- NODATA error message - The NetScaler sends a NODATA error message, if the domain name in query is valid but records of the given type are not available.

The NetScaler supports recursive resolution of DNS requests. In recursive resolution, the resolver (DNS client) sends a recursive query to a name server for a domain name. If the queried name server is authoritative for the domain, it responds with the requested domain name. Otherwise, the NetScaler queries the name servers recursively until the requested domain name is found.

Before you can apply the recursive query option, you must first enable it. You can also set the number of times the DNS resolver must send a resolution request (DNS retries) if a DNS lookup fails.

You can configure the NetScaler as a DNS forwarder. A forwarder passes DNS requests to external name servers. The NetScaler allows you to add external name servers and provides name resolution for domains outside the network. The NetScaler also allows you to set the name lookup priority to DNS or Windows Internet Name Service (WINS).

How DNS Works on the NetScaler

You can configure the NetScaler appliance to function as an ADNS server, DNS proxy server, end resolver, and forwarder. You can add DNS resource records on the NetScaler, including service (SRV) records, IPv6 (AAAA) records, address (A) records, mail exchange (MX) records, canonical name (CNAME) records, pointer (PTR) records, and start of authority (SOA) records. Also, you can configure the NetScaler to load balance external DNS name servers.

The NetScaler can be configured as the authority for a domain. To do this, you add valid SOA and NS records for the domain.

An ADNS server is a DNS server that contains complete information about a zone.

To configure the NetScaler as an ADNS server for a zone, you must add an ADNS service, and then configure the zone. To do so, you add valid SOA and NS records for the domain. When a client sends a DNS request, the NetScaler appliance searches the configured resource records for the domain name. You can configure the ADNS service to be used with the NetScaler Global Server Load Balancing (GSLB) feature.

You can delegate a subdomain, by adding NS records for the subdomain to the zone of the parent domain. You can then make the NetScaler authoritative for the subdomain, by adding a "glue record" for each of the subdomain name servers. If GSLB is configured, the NetScaler makes a GSLB load balancing decision based on its configuration and replies with the IP address of the selected virtual server. The following figure shows the entities in an ADNS GSLB setup and a DNS proxy setup.

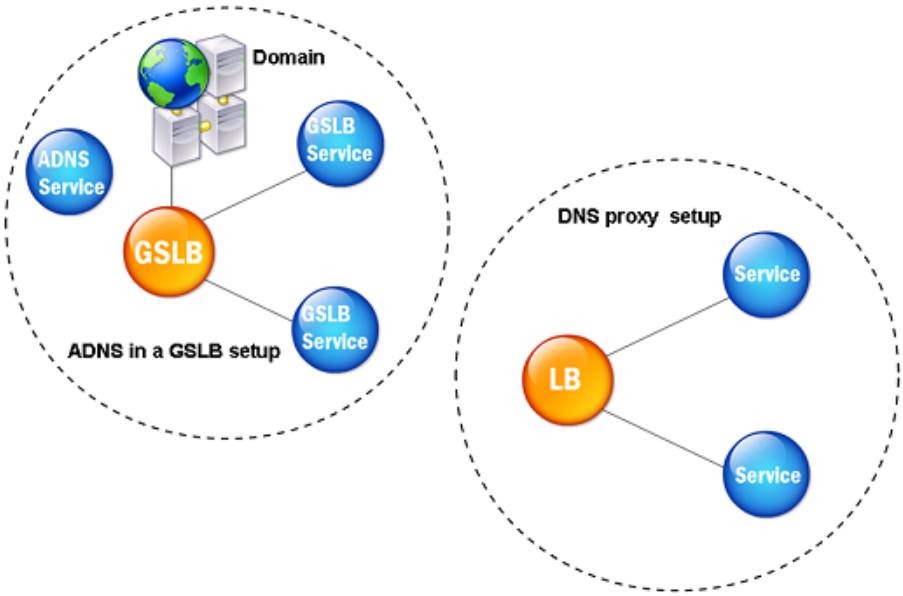


Figure 1. DNS Proxy Entity Model

The NetScaler can cache DNS responses (records) and can function as a DNS proxy. This enables the NetScaler to provide quick responses for repeated translations.

To configure the NetScaler as a DNS proxy, you must enable caching of DNS records. You must also create a load balancing DNS virtual server, and DNS services, and then bind these services to the virtual server.

The NetScaler provides two options, minimum time to live (TTL) and maximum TTL for configuring the lifetime of the cached data. The cached data times out as specified by your settings for these two options. The NetScaler checks the TTL of the DNS record coming from the server. If the TTL is less than the configured minimum TTL, it is replaced with the configured minimum TTL. If the TTL is greater than the configured maximum TTL, it is replaced with the configured maximum TTL.

The NetScaler also allows caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called *negative caching*. Negative caching helps speed up responses to queries on a domain, and can optionally provide the record type.

A negative response can be one of the following:

- NXDOMAIN error message - If a negative response is present in the local cache, the NetScaler returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the NetScaler. The NetScaler caches the response locally, then returns the error message to the client.

- NODATA error message - The NetScaler sends a NODATA error message, if the domain name in query is valid but records of the given type are not available.

The NetScaler supports recursive resolution of DNS requests. In recursive resolution, the resolver (DNS client) sends a recursive query to a name server for a domain name. If the queried name server is authoritative for the domain, it responds with the requested domain name. Otherwise, the NetScaler queries the name servers recursively until the requested domain name is found.

Before you can apply the recursive query option, you must first enable it. You can also set the number of times the DNS resolver must send a resolution request (DNS retries) if a DNS lookup fails.

You can configure the NetScaler as a DNS forwarder. A forwarder passes DNS requests to external name servers. The NetScaler allows you to add external name servers and provides name resolution for domains outside the network. The NetScaler also allows you to set the name lookup priority to DNS or Windows Internet Name Service (WINS).

Round Robin DNS

When a client sends a DNS request to find the DNS resource record, it receives a list of IP addresses resolving to the name in the DNS request. The client then uses one of the IP addresses in the list, generally, the first record or IP address. Hence, a single server is used for the total TTL of the cache and is overloaded when a large number of requests arrive.

When the NetScaler receives a DNS request, it responds by changing the order of the list of DNS resource records in a round robin method. This feature is called *round robin DNS*. Round robin distributes the traffic equally between data centers. The NetScaler performs this function automatically. You do not have to configure this behavior.

Functional Overview

If the NetScaler is configured as an ADNS server, it returns the DNS records in the order in which the records are configured. If the NetScaler is configured as a DNS proxy, it returns the DNS records in the order in which it receives the records from the server. The order of the records present in the cache matches the order in which records are received from the server.

The NetScaler then changes the order in which records are sent in the DNS response in a round robin method. The first response contains the first record in sequence, the second response contains the second record in sequence, the third response contains the third record in sequence, and the order continues in the same sequence. Thus, clients requesting the same name can connect to different IP addresses.

Round Robin DNS Example

As an example of round robin DNS, consider DNS records that have been added as follows:

```
add dns addRec ns1 1.1.1.1
add dns addRec ns1 1.1.1.2
add dns addRec ns1 1.1.1.3
add dns addRec ns1 1.1.1.4
```

The domain, abc.com is linked to an NS record as follows:

```
add dns nsrec abc.com. ns1
```

When the NetScaler receives a query for the A record of ns1, the Address records are served in a round robin method as follows. In the first DNS response, 1.1.1.1 is served as the first record:

```
ns1.            1H IN A      1.1.1.1
```

```
ns1.        1H IN A     1.1.1.2
ns1.        1H IN A     1.1.1.3
ns1.        1H IN A     1.1.1.4
```

In the second DNS response, the second IP address, 1.1.1.2 is served as the first record:

```
ns1.        1H IN A     1.1.1.2
ns1.        1H IN A     1.1.1.3
ns1.        1H IN A     1.1.1.4
ns1.        1H IN A     1.1.1.1
```

In the third DNS response, the third IP address, 1.1.1.2 is served as the first record:

```
ns1.        1H IN A     1.1.1.3
ns1.        1H IN A     1.1.1.4
ns1.        1H IN A     1.1.1.1
ns1.        1H IN A     1.1.1.2
```

Configuring DNS Resource Records

You configure resource records on the Citrix® NetScaler® appliance when you configure the appliance as an ADNS server for a zone. You can also configure resource records on the appliance if the resource records belong to a zone for which the appliance is a DNS proxy server. On the appliance, you can configure the following record types:

- Service records
- AAAA records
- Address records
- Mail Exchange records
- Name Server records
- Canonical records
- Pointer records
- Start of Authority records

The following table lists the record types and the number of records (per record type) that you can configure for a domain on the NetScaler.

Table 1. Record Type and Number Configurable

Record Type	Number of Records
Address (A)	25
IPv6 (AAAA)	5
Mail exchange (MX)	12
Name server (NS)	16
Service (SRV)	8
Pointer (PRT)	20
Canonical name (CNAME)	1
Start of Authority (SOA)	1

Creating SRV Records for a Service

The SRV record provides information about the services available on the NetScaler appliance. An SRV record contains the following information: name of the service and the protocol, domain name, TTL, DNS class, priority of the target, weight of records with the same priority, port of the service, and host name of the service. The NetScaler chooses the SRV record that has the lowest priority setting first. If a service has multiple SRV records with the same priority, clients use the weight field to determine which host to use.

To add an SRV record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an SRV record and verify the configuration:

- add dns srvRec <domain> <target> **-priority** <positive_integer> **-weight** <positive_integer> **-port** <positive_integer> [**-TTL** <secs>]
- sh dns srvRec <domain>

Example

```
> add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -weight 1 -port 80
 Done
> show dns srvRec _http._tcp.example.com
1)      Domain Name : _http._tcp.example.com
        Target Host : nameserver1.com
        Priority : 1    Weight : 1
        Port : 80       TTL : 5 secs <man page says 3600 secs>
 Done
>
```

To modify or remove an SRV record by using the NetScaler command line

- To modify an SRV record, type the set dns srvRec command, the name of the domain for which the SRV record is configured, the name of the target host that hosts the associated service, and the parameters to be changed, with their new values.
- To remove an SRV record, type the rm dns srvRec command, the name of the domain for which the SRV record is configured, and the name of the target host that hosts the associated service.

Parameters for configuring an SRV record

domain

The domain name that is offering the services. The domain name includes the service offered and the transport layer protocol (for example, `_ftp._tcp.abc.com`). This is a mandatory argument. Maximum length: 255.

target

The host for the specified service. This is a mandatory argument. Maximum length: 255.

priority

The priority that is assigned to the target host. The lower the priority value, the higher the priority. Clients always attempt to use the SRV record that has the lowest priority vlaue. This is a mandatory argument. Minimum value: 0. Maximum value: 65535.

weight

Weight for the target host. If two records have the same priority, the NetScaler selects the server based on the value of this parameter. This is a mandatory argument. Minimum value: 0. Maximum value: 65535.

port

The port name on which the target host is listening for client requests. This is a mandatory argument. Minimum value: 0. Maximum value: 65535.

TTL

The time to live, measured in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To configure an SRV record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **SRV Records**.
2. In the details pane, do one of the following:
   - To create a new SRV record, click **Add**.
   - To modify an existing SRV record, select the SRV record, and then click **Open**.
3. In the **Create SRV Record** or **Configure SRV Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SRV record" as shown:
   - **Domain Name*** —domain (cannot be changed for an existing SRV record)
   - **Target*** —target (cannot be changed for an existing SRV record)
   - **Priority*** —priority
   - **Weight*** —weight
   - **Port*** —port
   - **TTL** —TTL

   * A required parameter
4. Click **Create** or **OK**.

Creating AAAA Records for a Domain Name

An AAAA resource record stores a single IPv6 address.

To add an AAAA record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an AAAA record and verify the configuration:

- add dns aaaaRec <hostName> <IPv6Address> ... [**-TTL** <secs>]
- show dns aaaaRec <hostName>

Example

```
> add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57ab
 Done
> show dns aaaaRec www.example.com
1)      Host Name : www.example.com
       Record Type : ADNS            TTL : 5 secs
       IPV6 Address : 2001:db8::1428:57ab
 Done
>
```

To remove an AAAA record and all of the IPv6 addresses associated with the domain name, type the rm dns aaaaRec command and the domain name for which the AAAA record is configured. To remove only a subset of the IPv6 addresses associated with the domain name in an AAAA record, type the rm dns aaaaRec command, the domain name for which the AAAA record is configured, and the IPv6 addresses that you want to remove.

Parameters for configuring an AAAA record

hostName

The domain name for which the Address record is added. This is a mandatory argument. Maximum length: 255.

IPv6Address

The IPv6 address of the domain name.

TTL

The time to live, measured in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To add an AAAA record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **AAAA Records**.
2. In the details pane, click **Add**.
3. In the **Create AAAA Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an AAAA record" as shown:
   - **Host Name***—hostName
   - **IPv6 Address***—IPv6Address
   - **TTL**-TTL

   * A required parameter
4. Click **Add**. The IPv6 address appears in the **IP** box.
5. Click **Create**, and then click **Close**.

Creating Address records for a Domain Name

Address (A) records are DNS records that map a domain name to an IPv4 address.

You cannot delete Address records for a host participating in global server load balancing (GSLB). However, the NetScaler deletes Address records added for GSLB domains when you unbind the domain from a GSLB virtual server. Only user-configured records can be deleted manually. You cannot delete a record for a host referenced by records such as NS, MX, or CNAME.

To add an Address record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an Address record and verify the configuration:

- add dns addRec <hostName> <IPAddress> [**-TTL** <secs>]
- show dns addRec <hostName>

Example

```
> add dns addRec ns.example.com 192.0.2.0
 Done
> show dns addRec ns.example.com
1)      Host Name : ns.example.com
       Record Type : ADNS                  TTL : 5 secs
       IP Address : 192.0.2.0
 Done
>
```

To remove an Address record and all of the IP addresses associated with the domain name, type the rm dns addRec command and the domain name for which the Address record is configured. To remove only a subset of the IP addresses associated with the domain name in an Address record, type the rm dns addRec command, the domain name for which the Address record is configured, and the IP addresses that you want to remove.

Parameters for configuring an Address records

hostName

The domain name for which the Address record is being added. This is a mandatory argument. Maximum length: 255.

IPAddress

The IP address of the domain name.

TTL

The time to live, measured in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To add an Address record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **Address Records**.
2. In the details pane, click **Add**.
3. In the **Create Address Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring Address records" as shown:
   - **Host Name*** —hostName
   - **IP Address*** —IPAddress
   - **TTL**—TTL

   * A required parameter
4. Click **Add**. The IP address appears in the **IP Address** box.
5. Click **Create**, and then click **Close**.

Creating MX Records for a Mail Exchange Server

Mail Exchange (MX) records are used to direct email messages across the Internet. An MX record contains an MX preference that specifies the MX server to be used. The MX preference values range from 0 through 65536. An MX record contains a unique MX preference number. You can set the MX preference and the TTL values for an MX record.

When an email message is sent through the Internet, a mail transfer agent sends a DNS query requesting the MX record for the domain name. This query returns a list of host names of mail exchange servers for the domain, along with a preference number. If there are no MX records, the request is made for the Address record of that domain. A single domain can have multiple mail exchange servers.

To add an MX record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an MX record and verify the configuration:

- add dns mxRec <domain> **-mx** <string> **-pref** <positive_integer> [**-TTL** <secs>]
- show dns mxRec <domain>

Example

```
> add dns mxRec example.com -mx mail.example.com -pref 1
 Done
> show dns mxRec example.com
1)      Domain : example.com    MX Name : mail.example.com
        Preference :    1       TTL : 5 secs
 Done
>
```

To modify or remove an MX record by using the NetScaler command line

- To modify an MX record, type the set dns mxRec command, the name of the domain for which the MX record is configured, the name of the MX record, and the parameters to be changed, with their new values.
- To remove an an MX record, type the rm dns mxRec command, the name of the domain for which the MX record is configured, and the name of the MX record.

Parameters for configuring an MX record

domain

The domain for which the MX record is added. This is a mandatory argument. Maximum length: 255.

mx

The MX record name. This is a mandatory argument. Maximum length: 255.

pref

The route priority number. This is a mandatory argument. Minimum value: 0. Maximum value: 65535.

Note: A domain name can have multiple mail routes, with a priority number assigned to each. The mail route with the lowest number identifies the server responsible for the domain. Other listed mail servers are used as backups.

TTL

The time to live, in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To add an MX record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **Mail Exchange Records**.

2. In the details pane, do one of the following:

   - To create a new MX record, click **Add**.

   - To modify an existing MX record, select the MX record, and then click **Open**.

3. In the **Create Mail Exchange Record** or **Configure Mail Exchange Record** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring an MX record” as shown:

   - **Domain Name***—domain (cannot be changed for an existing MX record)

   - **Mail Exchange***—mx (cannot be changed for an existing MX record)

   - **Preference No.***—pref

   - **TTL**—TTL

   * A required parameter

4. Click **Create** or **OK**.

Creating NS Records for an Authoritative Server

Name Server (NS) records specify the authoritative server for a domain. You can configure a maximum of 16 NS records. You can use an NS record to delegate the control of a subdomain to a DNS server.

To create an NS record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an NS record and verify the configuration:

- add dns nsRec <domain> <nameServer> [**-TTL** <secs>]
- show dns nsRec <domain>

Example

```
> add dns nsRec example.com nameserver1.example.com
 Done
> show dns nsRec example.com
1)      Domain : example.com    NameServer : nameserver1.example.com
       TTL : 5 sec
 Done
>
```

To remove an NS record, type the rm dns nsRec command, the name of the domain to which the NS record belongs, and the name of the name server.

Parameters for configuring an NS record

domain

The domain name for which the name server record is being added.

nameServer

The name server for the domain.

TTL

The time to live, measured in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To create an NS record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **Name Server Records**.
2. In the details pane, click **Add**.
3. In the **Create Name Server Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an NS record" as shown:
   - **Domain Name*** —domain
   - **Name Server*** —nameServer
   - **TTL** —TTL

   * A required parameter
4. Click **Create**, and then click **Close**.

Creating CNAME Records for a Subdomain

A canonical name record (CNAME record) is an alias for a DNS name. These records are useful when multiple services query the DNS server. The host that has an address (A) record cannot have a CNAME record.

To add a CNAME record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a CNAME record and verify the configuration:

- add dns cnameRec <aliasName> <canonicalName> [**-TTL** <secs>]
- show dns cnameRec <aliasName>

Example

```
> add dns cnameRec www.example.com www.examplenw.com
 Done
> show dns cnameRec www.example.com
       Alias Name      Canonical Name  TTL
1)      www.example.com        www.examplenw.com       5 secs
 Done
>
```

To remove a CNAME record for a given domain, type the rm dns cnameRec command and the alias of the domain name.

Parameters for configuring a CNAME record

aliasName

Domain name for the defined alias. Maximum length: 256.

canonicalName

Alias name for the specified domain. Maximum length: 256.

TTL

The time to live, measured in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To add a CNAME record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **Canonical Records**.
2. In the details pane, click **Add**.
3. In the **Create Canonical Name Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a CNAME record" as shown:
   - **Alias Name*** —aliasName
   - **Canonical Name*** —canonicalName
   - **TTL**—TTL

   * A required parameter
4. Click **Create**, and then click **Close**.

Creating PTR Records for IPv4 and IPv6 Address

A pointer (PTR) record translates an IP address to its domain name. IPv4 PTR records are represented by the octets of an IP address in reverse order with the string "in-addr.arpa." appended at the end. For example, the PTR record for the IP address 1.2.3.4 is 4.3.2.1.in-addr.arpa.

IPv6 addresses are reverse mapped under the domain IP6.ARPA. IPv6 reverse-maps use a sequence of nibbles separated by dots with the suffix ".IP6.ARPA" as defined in RFC 3596. For example, the reverse lookup domain name corresponding to the address, 4321:0:1:2:3:4:567:89ab would be b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.

To add a PTR record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a PTR record and verify the configuration:

- add dns ptrRec <reverseDomain> <domain> [**-TTL** <secs>]
- show dns ptrRec <reverseDomain>

Example

```
> add dns ptrRec 0.2.0.192.in-addr.arpa example.com
 Done
> show dns ptrRec 0.2.0.192.in-addr.arpa
1)      Reverse Domain Name : 0.2.0.192.in-addr.arpa
       Domain Name : example.com              TTL : 3600 secs
 Done
>
```

To remove a PTR record, type the rm dns ptrRec command and the reverse domain name associated with the PTR record

Parameters for configuring a PTR record

reverseDomain

Reversed representation of the domain name that the PTR record must point to. Possible suffix values are in-addr.arpa. for IPv4 addresses and ip6.arpa. for IPv6 addresses. This is a mandatory argument. Maximum length: 75.

domain

The domain name for which reverse mapping is being done. This is a mandatory argument. Maximum length: 255.

TTL

The time to live, measured in seconds. Minimum value: 0. Maximum value: 2147483647. Default: 3600.

To add a PTR record by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **PTR Records**.
2. In the details pane, click **Add**.
3. In the **Create PTR Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a PTR record" as shown:
   - **Choose suffix***—(Select **.in-addr.arpa.** to specify a PTR record for an IPv4 address or **.ip6.arpa.** to specify a PTR record for an IPv6 address. The string that you select is appended to the reversed IP address to form the reverse domain name.)
   - **IP Address***—(The IP address of the domain name. The **Reverse Domain Name** box displays the reverse domain name that is generated when you are entering the IP address. After you enter the IP address, verify that the reverse domain name is correct.)
   - **Domain***—domain
   - **TTL**—TTL

   * A required parameter
4. Click **Add**.

   The domain name appears in the **Domain** list. Add as many domain names as you want to the **Domain** list.
5. Click **Create**, and then click **Close**.

Creating SOA Records for Authoritative Information

A Start of Authority (SOA) record is created only at the zone apex and contains information about the zone. The record includes, among other parameters, the primary name server, contact information (e-mail), and default (minimum) time-to-live (TTL) values for records.

To create an SOA record by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an SOA record and verify the configuration:

- add dns soaRec <domain> **-originServer** <originServerName> **-contact** <contactName>
- sh dns soaRec <do main>

Example

```
> add dns soaRec example.com -originServer nameserver1.example.com -contact admin.example.com
 Done
> show dns soaRec example.com
1)      Domain Name : example.com
        Origin Server : nameserver1.example.com
        Contact : admin.example.com
        Serial No. : 100        Refresh : 3600 secs     Retry : 3 secs
        Expire : 3600 secs      Minimum : 5 secs        TTL : 3600 secs
 Done
>
```

To modify or remove an SOA record by using the NetScaler command line

- To modify an SOA record, type the set dns soaRec command, the name of the domain for which the record is configured, and the parameters to be changed, with their new values.
- To remove an SOA record, type the rm dns soaRec command and the name of the domain for which the record is configured.

Parameters for configuring an SOA record

domain

Domain name for which the SOA record is added.

originServer

Name of the origin server for the given domain.

contact

Contact person for this ADNS server. This is typically an e-mail address in which the at sign (@) is replaced by a period (.).

To configure an SOA record by using the configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **SOA Records**.
2. In the details pane, do one of the following:
   - To create an SOA record, click **Add**.
   - To configure an existing SOA record, select the SOA record, and then click **Open**.
3. In the **Create SOA Record** or **Configure SOA Record** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SOA record" as shown:
   - **Domain Name***—domain
   - **Origin Server***—originServer
   - **Contact***—contact

   * A required parameter
4. Click **Create**, and then click **Close**.

Viewing DNS Statistics

You can view the DNS statistics generated by the Citrix® NetScaler® appliance. The DNS statistics include runtime, configuration, and error statistics.

To view DNS records statistics by using the NetScaler command line

At the NetScaler command prompt, type:

stat dns

Example

```
> stat dns
DNS Statistics

Runtime Statistics
Dns queries                    21
NS queries                      8
SOA queries                    18
.
.
.
Configuration Statistics
AAAA records                   17
A records                      36
MX records                      9
.
.
.
Error Statistics
Nonexistent domain             17
No AAAA records                 0
No A records                   13
.
.
.
Done
>
```

To view DNS records statistics by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, click **Statistics**.

Configuring a DNS Zone

A zone entity on the Citrix® NetScaler® appliance facilitates the ownership of a domain on the appliance. It is also used in the context of DNS Security Extensions (DNSSEC). DNSSEC sign operations are performed on all the resource records in a DNS zone. Therefore, if you want to sign a zone, you must first create the zone on the NetScaler appliance.

You must create a DNS zone in the following scenarios:

- The NetScaler appliance owns all the records in a zone, that is, the appliance is operating as the authoritative DNS server for the zone. The zone must be created with the proxyMode parameter set to NO.
- The NetScaler appliance owns only a subset of the records in a zone, and all the other resource records in the zone are hosted on a set of back-end name servers for which the appliance is configured as a DNS proxy server. A typical configuration where the NetScaler appliance owns only a subset of the resource records in the zone is a global server load balancing (GSLB) configuration. Only the GSLB domain names are owned by the NetScaler appliance, while all the other records are owned by the back-end name servers. You must set the proxyMode parameter to YES.

If the NetScaler appliance does not host any of the resource records in a zone, you need not create a zone on the appliance.

Note: If the NetScaler is operating as the authoritative DNS server for a zone, you must create Start of Authority (SOA) and name server (NS) records for the zone before you create the zone. If the NetScaler is operating as the DNS proxy server for a zone, SOA and NS records must not be created on the NetScaler appliance. For more information about creating SOA and NS records, see Configuring DNS Resource Records.

When you create a zone, all existing domain names and resource records that end with the name of the zone are automatically treated as a part of the zone. Additionally, any new resource records created with a suffix that matches the name of the zone are implicitly included in the zone.

To create a DNS zone on the NetScaler appliance by using the NetScaler command line

At the NetScaler command prompt, type the following command to add a DNS zone to the NetScaler appliance and verify the configuration:

- add dns zone <zoneName> **-proxyMode** (YES | NO)
- show dns zone [<zoneName> | **-type** <type>]

Example

```
> add dns zone example.com -proxyMode Yes
 Done
> show dns zone example.com
        Zone Name : example.com
        Proxy Mode : YES
Done
>
```

To modify or remove a DNS zone by using the NetScaler command line

- To modify a DNS zone, type the set dns zone command, the name of the DNS zone, and the parameters to be changed, with their new values.
- To remove a DNS zone, type the rm dns zone command and the name of the dns zone.

Parameters for configuring a DNS zone

zoneName

The name of the zone being added. This is a mandatory argument. Maximum length: 255

proxyMode

Specifies whether the zone is deployed in proxy mode. This is a mandatory argument. Possible values: YES, NO. Default value: ENABLED

To configure a DNS zone by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Zones**.
2. In the details pane, do one of the following:
   - To create a DNS zone, click **Add**.
   - To modify an existing DNS zone, select the zone, and then click **Open**.
3. In the **Create DNS Zone** or **Configure DNS Zone** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a DNS zone," as shown:
   - **DNS Zone*** —zoneName (cannot be changed for an existing DNS zone)
   - **Proxy Mode**—proxyMode

   * A required parameter
4. Click **Create** or **OK**.
5. In the details pane, click the name of the zone you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring the NetScaler as an ADNS Server

You can configure the Citrix® NetScaler® appliance to function as an authoritative domain name server (ADNS) for a domain. As an ADNS server for a domain, the NetScaler resolves DNS requests for all types of DNS records that belong to the domain. To configure the NetScaler to function as an ADNS server for a domain, you must create an ADNS service and configure NS and Address records for the domain on the NetScaler. Normally, the ADNS service uses the Mapped IP address (MIP). However, you can configure the ADNS service with any NetScaler-owned IP address. The following topology diagram shows a sample configuration and the flow of requests and responses.

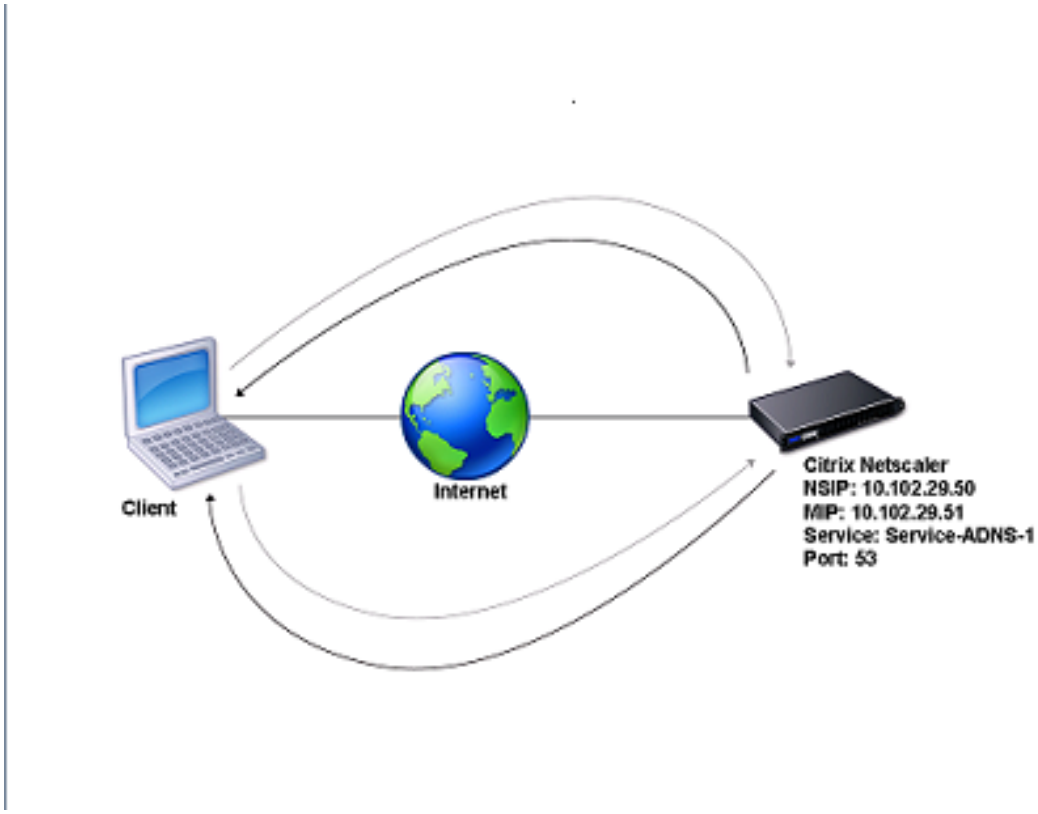


Figure 1. NetScaler as an ADNS

The following table shows the parameters that are configured for the ADNS service illustrated in the preceding topology diagram.

Table 1. Example of ADNS Service Configuration

Entity type	Name	IP address	Type	Port
ADNS Service	Service-ADNS-1	10.102.29.51	ADNS	53

To configure an ADNS setup, you must configure the ADNS service. For instructions on configuring the ADNS service, see Load Balancing.

During DNS resolution, the ADNS server directs the DNS proxy or local DNS server to query the NetScaler for the IP address of the domain. Because the NetScaler is authoritative for the domain, it sends the IP address to the DNS proxy or local DNS server. The following diagram describes the placement and role of the ADNS server in a GSLB configuration.

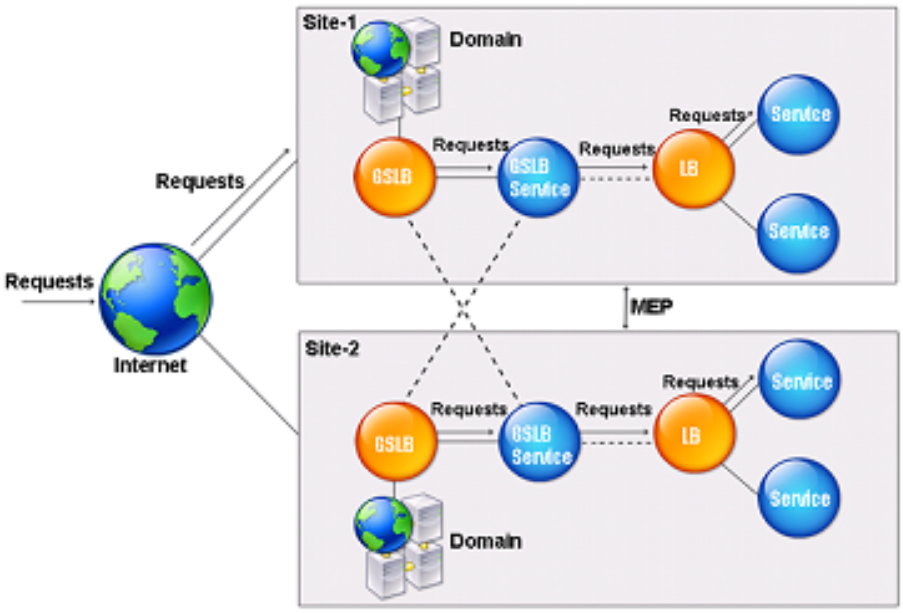


Figure 2. GSLB Entity Model

Note: In ADNS mode, if you remove SOA and ADNS records, the following do not function for the domain hosted by the NetScaler: ANY query (for more information about the ANY query, see DNS ANY Query), and negative responses, such as NODATA and NXDOMAIN.

Creating an ADNS Service

An ADNS service is used for global service load balancing. For more information about creating a GSLB setup, see Global Server Load Balancing. You can add, modify, enable, disable, and remove an ADNS service. For instructions on creating an ADNS service, see Configuring Services. For examples of the parameter values, see the table on page 521.

Note: You can configure the ADNS service to use MIP, SNIP, or any new IP address.

When you create an ADNS service, the NetScaler responds to DNS queries on the configured ADNS service IP and port.

You can verify the configuration by viewing the properties of the ADNS service You can view properties such as name, state, IP address, port, protocol, and maximum client connections. For instructions on viewing the properties of an ADNS service, see Viewing the Properties of a Service.

Configuring the ADNS Setup to Use TCP

By default, some clients use the User Datagram Protocol (UDP) for DNS, which specifies a limit of 512 bytes for the payload length of UDP packets. To handle payloads that exceed 512 bytes in size, the client must use the Transmission Control Protocol (TCP). To enable DNS communications over TCP, you must configure the NetScaler appliance to use the TCP protocol for DNS. The NetScaler then sets the truncation bit in the DNS response packets. The truncation bit specifies that the response is too large for UDP and that the client must send the request over a TCP connection. The client then uses the TCP protocol on port 53 and opens a new connection to the NetScaler. The NetScaler listens on port 53 with the IP address of the ADNS service to accept the new TCP connections from the client.

To configure the NetScaler to use the TCP protocol, you must configure an ADNS_TCP service. For instructions on creating an ADNS_TCP service, see Load Balancing.

Important: To configure the NetScaler to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure the ADNS and ADNS_TCP services. The IP address of the ADNS_TCP service must be same as the IP address of the ADNS service.

Adding DNS Resource Records

After you create an ADNS service, you can add DNS records. For instructions on adding DNS records, see Configuring DNS Resource Records.

Removing ADNS Services

For instructions on removing services, see Load Balancing.

Configuring Domain Delegation

Domain delegation is the process of assigning responsibility for a part of the domain space to another name server. Therefore, during domain delegation, the responsibility for responding to the query is delegated to another DNS server. Delegation uses NS records.

In the following example, sub1.abc.com is the subdomain for abc.com. The procedure describes the steps to delegate the subdomain to the name server ns2.sub1.abc.com and add an Address record for ns2.sub1.abc.com.

To configure domain delegation, you need to perform the following tasks, which are described in the sections that follow:

1. Create an SOA record for a domain.
2. Create an NS record to add a name server for the domain.
3. Create an Address record for the name server.
4. Create an NS record to delegate the subdomain.
5. Create a glue record for the name server.

Creating an SOA Record

For instructions on configuring SOA records, see Creating SOA Records for Authoritative Information.

Creating an NS Record for a Name Server

For instructions on configuring an NS record, see Creating NS Records for an Authoritative Server. In the **Name Server** drop-down list, select the primary authoritative name server, for example, ns1.abc.com.

Creating an Address Record

For instructions on configuring Address records, see Creating Address Records for a Domain Name. In the **Host Name** and **IP address** text boxes, type the domain name for the DNS Address record and the IP address, for example, ns1.abc.com and 10.102.11.135, respectively.

Creating an NS Record for Domain Delegation

For instructions on configuring NS records, see Creating NS Records for an Authoritative Server. In the **Name Server** drop-down list, select the primary authoritative name server, for example, ns2.sub1.abc.com.

Creating a Glue Record

NS records are usually defined immediately after the SOA record (but this is not a restriction.) A domain must have at least two NS records. If an NS record is defined within a domain, it must have a matching Address record. This Address record is referred to as a glue record. Glue records speed up DNS queries.

For instructions on adding glue records for a subdomain, see the procedure for adding an Address (A) record, Configuring DNS Resource Records.

For instructions on configuring Address records, see Creating Address Records for a Domain Name. In **Host Name** and **IP address** text boxes, type the domain name for the DNS Address record and the IP address, for example, ns2.sub1.abc.com and 10.102.12.135, respectively.

Configuring the NetScaler as a DNS Proxy Server

As a DNS proxy server, the Citrix® NetScaler® appliance can function as a proxy for either a single DNS server or a group of DNS servers. The flow of requests and responses is illustrated in the following sample topology diagram.

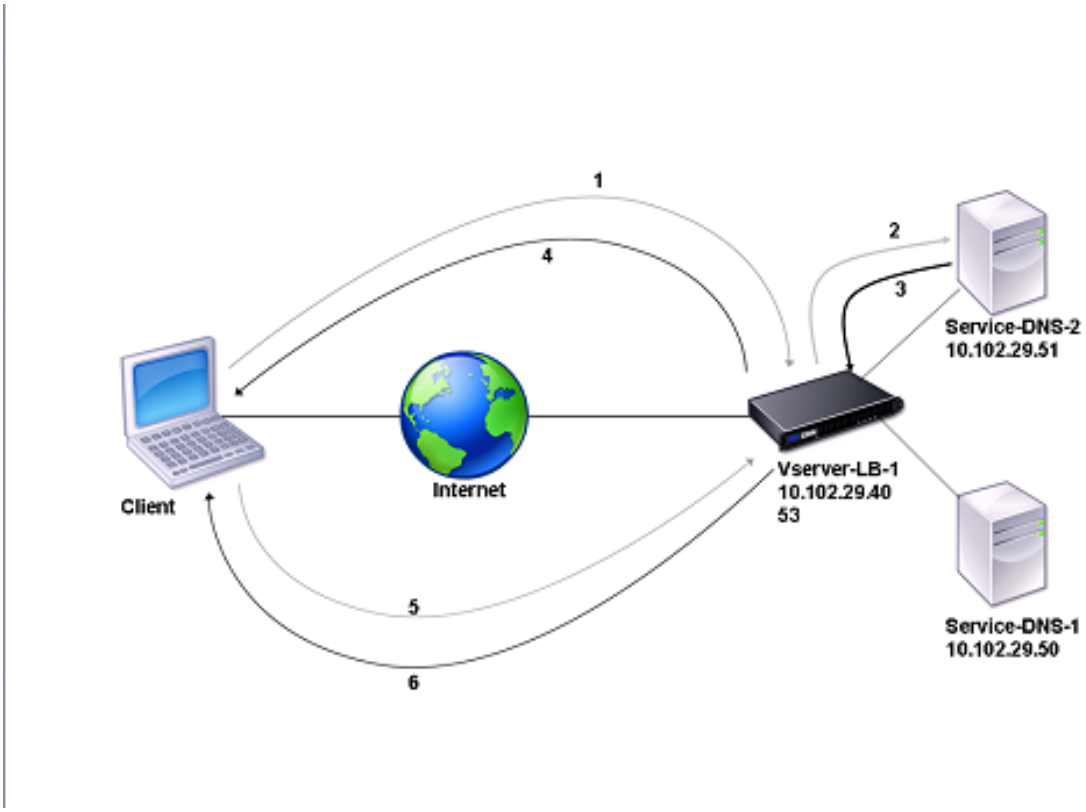


Figure 1. NetScaler as DNS proxy

To resolve a domain name into its IP address, the NetScaler checks for the queried domain in its local cache. If the address for the queried domain is present in the local cache, the NetScaler returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the NetScaler. The NetScaler then returns the address to the client.

For requests for a domain that has been cached earlier, the NetScaler serves the Address record of the domain from the cache without querying the configured DNS server.

The NetScaler discards a record stored in its cache when the time-to-live (TTL) value of the record reaches the configured value. A client that requests an expired record has to wait until the NetScaler retrieves the record from the server and updates its cache. To avoid this delay, the NetScaler proactively updates the cache by retrieving the record from the server before the record expires.

The following table lists sample names and the values of the entities that need to be configured on the NetScaler.

Table 1. Example of DNS Proxy Entity Configuration

Entity type	Name	IP address	Type	Port
LB virtual server	Vserver-DNS-1	10.102.29.40	DNS	53
Services	Service-DNS-1	10.102.29.50	DNS	53
	Service-DNS-2	10.102.29.51	DNS	53

The following diagram shows the entities of a DNS Proxy and the values of the parameters to be configured on the NetScaler.

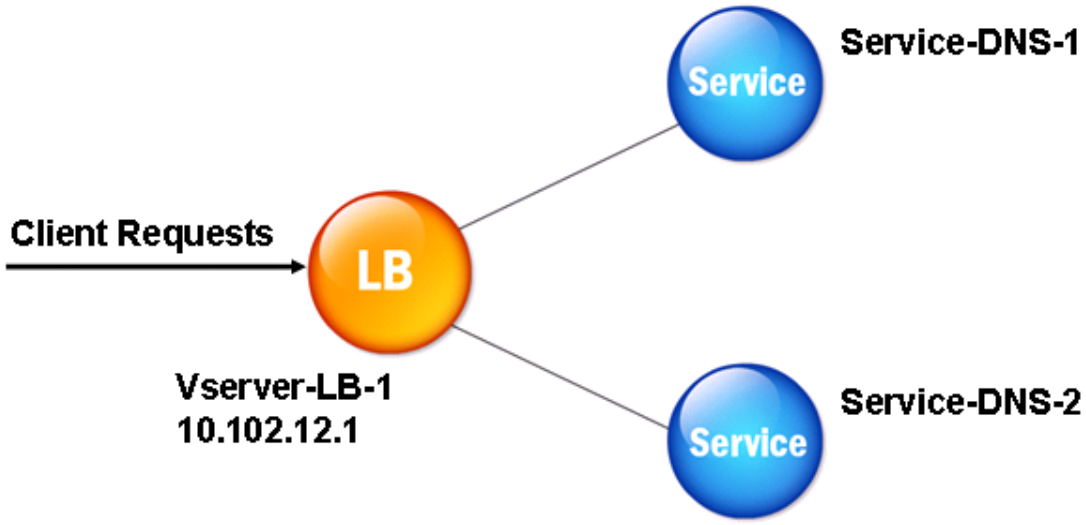


Figure 2. DNS Proxy Entity Model

Note: To configure DNS proxy, you need to know how to configure load balancing services and virtual servers. For information about configuring load balancing services and virtual servers, see Load Balancing, and then configure DNS proxy setup.

Creating a Load Balancing Virtual Server

To configure a DNS Proxy on the NetScaler, you must configure a load balancing virtual server of type DNS. You can add, modify, enable, disable, and remove load balancing virtual servers. For instructions on creating a load balancing virtual server, see Load Balancing.

Creating DNS Services

After creating a load balancing virtual server of type DNS, you must create DNS services. You can add, modify, enable, disable, and remove a DNS service. For instructions on creating a DNS service, see Load Balancing.

Binding a Load Balancing Virtual Server to DNS Services

To complete the DNS Proxy configuration, you must bind the DNS services to the load balancing virtual server. For instructions on binding a service to a load balancing virtual server, see Load Balancing.

Configuring the DNS Proxy Setup to Use TCP

Some clients use the User Datagram Protocol (UDP) for DNS communications. However, UDP specifies a maximum packet size of 512 bytes. When payload lengths exceed 512 bytes, the client must use the Transmission Control Protocol (TCP). When a client sends the Citrix® NetScaler® appliance a DNS query, the appliance forwards the query to one of the name servers. If the response is too large for a UDP packet, the name server sets the truncation bit in its response to the NetScaler. The truncation bit indicates that the response is too large for UDP and that the client must send the query over a TCP connection. The NetScaler relays the response to the client with the truncation bit intact and waits for the client to initiate a TCP connection with the IP address of the DNS load balancing virtual server, on port 53. The client sends the request over a TCP connection. The NetScaler appliance then forwards the request to the name server and relays the response to the client.

To configure the NetScaler to use the TCP protocol for DNS, you must configure a load balancing virtual server and services, both of type DNS_TCP. You can configure monitors of type DNS_TCP to check the state of the services. For instructions on creating DNS_TCP virtual servers, services, and monitors, see Load Balancing.

For updating the records proactively, the NetScaler uses a TCP connection to the server to retrieve the records.

Important: To configure the NetScaler to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure DNS and DNS_TCP services. The IP address of the DNS_TCP service must be same as that of the DNS service.

Enabling Caching of DNS Records

To complete the process of configuring a DNS proxy on the NetScaler, you must enable caching of DNS records. You must also specify minimum and maximum time-to-live (TTL) values for the records that are cached. The TTL values are measured in seconds.

To enable caching of DNS records by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable caching of DNS records and verify the configuration:

- set dns parameter **-cacheRecords** Yes
- show dns parameter

Example

```
> set dns parameter -cacheRecords YES
 Done
> show dns parameter
          .
          .
          .
      Cache Records : YES
          .
          .
          .
 Done
>
```

To enable caching of DNS records by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, under **Settings**, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, select the **Enable records caching** check box, and then click **OK**.

Configuring Time-to-Live Values for DNS Entries

The TTL is the same for all DNS records with the same domain name and record type. If the TTL value is changed for one of the records, the new value is reflected in all records of the same domain name and type. The default TTL value is 3600 seconds. The minimum is 0, and the maximum is 2147483647. If a DNS entry has a TTL value less than the minimum or greater than the maximum, it is saved as the minimum or maximum TTL value, respectively.

To specify the minimum and/or maximum TTL by using the NetScaler command line

At the NetScaler command prompt, type the following commands to specify the minimum and maximum TTL and verify the configuration:

- set dns parameter [**-minTTL** <secs>] [**-maxTTL** <secs>]
- show dns parameter

Example

```
> set dns parameter -minTTL 1200 -maxTTL 1800
 Done
> show dns parameter
      DNS parameters:
      DNS retries: 5
      Minimum TTL: 1200            Maximum TTL: 1800
         .
         .
         .
 Done
>
```

To specify the minimum and/or maximum TTL by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, under **Settings**, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, in **TTL**, in the **Minimum and Maximum** text boxes, type the minimum and maximum time to live (in seconds), respectively, and then click **OK**.

Note: When the TTL expires, the record is deleted from the cache. The NetScaler proactively contacts the servers and obtains the DNS record just before the DNS record expires.

Flushing DNS Records

You can delete all DNS records present in the cache. For example, you might want to flush DNS records when a server is restarted after modifications are made.

To delete all proxy records by using the NetScaler command line

At the NetScaler command prompt, type:

flush dns proxyRecords

To delete all proxy records by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **Address Records**.
2. In the details pane, click **Flush Proxy Records**.

Adding DNS Resource Records

You can add DNS records to a domain for which the Citrix® NetScaler® appliance is configured as a DNS proxy server. For information about adding DNS records, see Configuring DNS Resource Records.

Removing a Load Balancing DNS Virtual Server

For information about removing a load balancing virtual server, see Load Balancing.

Limiting the Number of Concurrent DNS Requests on a Client Connection

You can limit the number of concurrent DNS requests on a single client connection, which is identified by the <clientip:port>-<vserver ip:port> tuple. Concurrent DNS requests are those requests that the NetScaler appliance has forwarded to the name servers and for which the appliance is awaiting responses. Limiting the number of concurrent requests on a client connection enables you to protect the name servers when a hostile client attempts a Distributed Denial of Service (DDoS) attack by sending a flood of DNS requests. When the limit for a client connection is reached, subsequent DNS requests on the connection are dropped till the outstanding request count goes below the limit. This limit does not apply to the requests that the NetScaler appliance serves out of its cache.

The default value for this parameter is 255. This default value is sufficient in most scenarios. If the name servers serve a large number of concurrent DNS requests under normal operating conditions, you can specify either a large value or a value of zero (0). A value of 0 disables this feature and specifies that there is no limit to the number of DNS requests that are allowed on a single client connection. This is a global parameter and applies to all the DNS virtual servers that are configured on the NetScaler appliance.

To specify the maximum number of concurrent DNS requests allowed on a single client connection by using the NetScaler command line

At the NetScaler command prompt, type the following commands to specify the maximum number of concurrent DNS requests allowed on a single client connection and verify the configuration:

- set dns parameter **-maxPipeline** <positive_integer>
- show dns parameter

Example

```
> set dns parameter -maxPipeline 1000
 Done
> show dns parameter
      DNS parameters:
      DNS retries: 5
      .
      .
```

```
        .
        Max DNS Pipeline Requests: 1000
 Done
>
```

Parameters for specifying the maximum number of concurrent DNS requests on a single client connection

maxPipeline

Specifies the maximum number of concurrent DNS requests that are allowed on a single client connection. A value of 0 (zero) implies that there is no limit to the number of concurrent DNS requests that are allowed on a single client connection. Default value: 255

To specify the maximum number of concurrent DNS requests allowed on a single client connection by using the configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, specify a value for **Max DNS Pipeline Requests**, which corresponds to the parameter described in "Parameters for specifying the maximum number of concurrent DNS requests on a single client connection."
4. Click **OK**.

Configuring the NetScaler as an End Resolver

A resolver is a procedure that is invoked by an application program that translates a domain/host name to its resource record. The resolver interacts with the LDNS, which looks up the domain name to obtain its IP address. The NetScaler can provide end-to-end resolution for DNS queries.

In recursive resolution, the NetScaler appliance queries different name servers recursively to access the IP address of a domain. When the NetScaler receives a DNS request, it checks its cache for the DNS record. If the record is not present in the cache, it queries the root servers configured in the ns.conf file. The root name server reports back with the address of a DNS server that has detailed information about the second-level domain. The process is repeated until the required record is found.

When you start the NetScaler appliance for the first time, 13 root name servers are added to the ns.conf file. The NS and Address records for the 13 root servers are also added. You can modify the ns.conf file, but the NetScaler does not allow you to delete all 13 records; at least one name server entry is required for the appliance to perform name resolution. The following diagram illustrates the process of of name resolution.

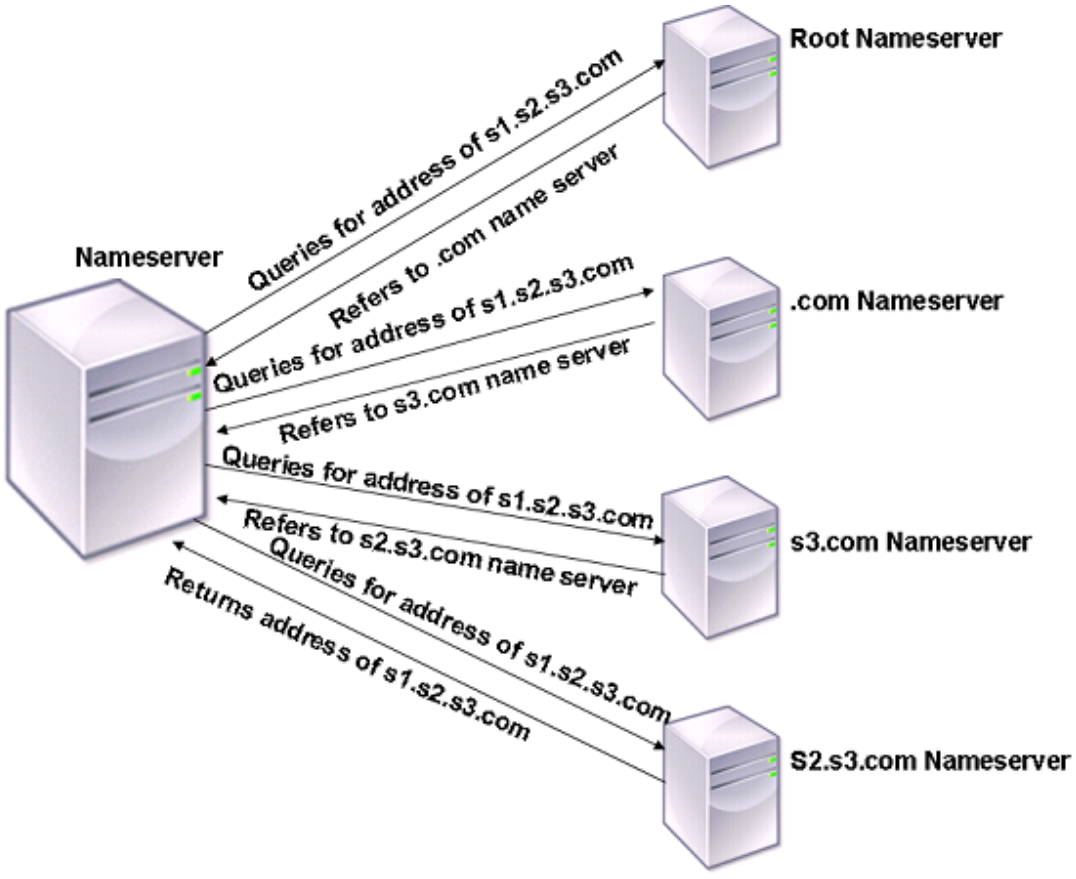


Figure 1. Recursive Resolution

In the process shown in the diagram, when the name server receives a query for the address of s1.s2.s3.com, it first checks the root name servers for s1.s2.s3.com. A root name server reports back with the address of the .com name server. If the address of s1.s2.s3.com is found in the name server, it responds with a suitable IP address. Otherwise, it queries other name servers for s3.com, then for s2.s3.com to retrieve the address of s1.s2.s3.com. In this way, resolution always starts from root name servers and ends with the domain's authoritative name server.

Note: For recursive resolution functionality, caching should be enabled.

Enabling Recursive Resolution

To configure the NetScaler appliance to function as an end resolver, you must enable recursive resolution on the appliance.

To enable recursive resolution by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable recursive resolution and verify the configuration:

- set dns parameter **-recursion** ENABLED
- show dns parameter

Example

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
      DNS parameters:
          .
          .
          .
      Recursive Resolution : ENABLED
          .
          .
          .
 Done
>
```

To enable recursive resolution by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, under **Settings**, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, select the **Enable recursion** check box, and then click **OK**.

Setting the Number of Retries

The NetScaler appliance can be configured to make a preconfigured number of attempts (called DNS retries) when it does not receive a response from the server to which it sends a query. By default, the number of DNS retries is set to 5.

To set the number of DNS retries by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the number of retries and verify the configuration:

- set dns parameter **-retries** <positive_integer>
- show dns parameter

Example

```
> set DNS parameter -retries 3
 Done
> show dns parameter
       DNS parameters:
       DNS retries: 3
            .
            .
            .
 Done
>
```

To set the number of retries by using the configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, under **Settings**, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, in the **DNS Retries** text box, type the DNS resolver request retry count, and then click **OK**.

Configuring the NetScaler as a Forwarder

A forwarder is a server that forwards DNS queries to DNS servers that are outside the forwarder server's network. Queries that cannot be resolved locally are forwarded to other DNS servers. A forwarder accumulates external DNS information in its cache as it resolves DNS queries. To configure the NetScaler as a forwarder, you must add an external name server (a name server other than the Citrix NetScaler appliance).

The NetScaler appliance allows you to add external name servers to which it can forward the name resolution queries that cannot be resolved locally. To configure the NetScaler appliance as a forwarder, you must add the name servers to which it should forward name resolution queries. You can specify the lookup priority to specify the name service that the NetScaler appliance must use for name resolution.

Adding a Name Server

You can create a name server by specifying its IP address or by configuring an existing virtual server as the name server.

While adding name servers, you can provide an IP address or a virtual IP address (VIP). If you add an IP address, the NetScaler load balances requests to the configured name servers in round robin method. If you add a VIP, you can configure any load balancing method.

Example 1, which follows the command synopsis below, adds a local name server. Example 2 specifies the name of a load balancing virtual server of service type DNS.

Note: To verify the configuration, cou can also use the `sh dns <recordtype> <domain>` command. If the queried records are not present in the cache, the resource records are fetched from the configured external name servers.

To add a name server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a name server and verify the configuration:

- add dns nameServer ((<IP> [**-local**]) | <dnsVserverName>)
- show dns nameServer [<IP> | <dnsVserverName>]

Example 1

```
> add dns nameServer 10.102.9.20 -local
 Done
> show dns nameServer 10.102.9.20
1)      10.102.9.20: LOCAL  -  State: UP
Done
>
```

Example 2

```
> add dns nameServer dnsVirtualNS
 Done
> show dns nameServer dnsVirtualNS
1)      dnsVirtualNS  -  State: DOWN
```

```
 Done
>
```

To remove a name server by using the NetScaler command line, at the NetScaler command prompt, type the rm dns nameServer command followed by the IP address of the name server.

Parameters for adding a name server

IP

The IP address of the name server.

local

Specifies that the IP address belongs to a local recursive name server.

dnsVserverName

The name of a DNS virtual server. Maximum length: 127.

To add a name server by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Name Servers**.
2. In the details pane, click **Add**.
3. In the **Create Name Server** dialog box, do one of the following:
   - To add an IP address, click **IP Address**, and in the **IP Address** text box, type the IP address of the name server, for example, **10.102.29.10**. If you are adding an external name server, clear the **Local** check box.
   - To add a DNS virtual server, click **DNS Virtual Server**, and select a DNS virtual server. Click **New** if you want to create a new load balancing virtual server. The **Create Virtual Server (Load Balancing)** dialog box appears.
4. Click **Create**, and then click **Close**.

Note: When name servers are added in the Forwarder mode, the **Local** option must be cleared. When name servers are added in the End Resolver mode, the **Local** option must be selected.

Setting DNS Lookup Priority

You can set the lookup priority to either DNS or WINS. This option is used in the SSL VPN mode of operation.

To set the lookup priority to DNS by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the lookup priority to DNS and verify the configuration:

- set dns parameter **-nameLookupPriority** (DNS | WINS)
- show dns parameter

Example

```
> set dns parameter -nameLookupPriority DNS
 Done
> show dns parameter
          .
          .
          .
       Name lookup priority : DNS
          .
          .
          .
 Done
>
```

To set lookup priority to DNS by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, under **Settings**, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, under **Name Lookup Priority**, select **DNS** or **WINS**, and then click **OK**.

Note: If the DNS virtual server that you have configured is DOWN and if you set the **-nameLookupPriority** to DNS, the NetScaler does not attempt WINS lookup. Therefore, if a DNS virtual server is not configured or is disabled, set the **-nameLookupPriority** to WINS.

Disabling and Enabling Name Servers

The following procedure describes the steps to enable or disable an existing name server.

To enable or disable a name server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable or disable a name server and verify the configuration:

- (enable | disable) dns nameServer <IPAddress>
- show dns nameServer <IPAddress>

Example

```
> disable dns nameServer 10.102.9.19
 Done
> show dns nameServer 10.102.9.19
1)      10.102.9.19: LOCAL  -  State: OUT OF SERVICE
 Done
>
```

To enable or disable a name server by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Name Servers**.
2. In the details pane, select the name server that you want to enable or disable.
3. Click **Enable** or **Disable**. If a name server is enabled, the **Disable** option is available. If a name server is disabled, the **Enable** option is available.

Configuring DNS Suffixes

You can configure DNS suffixes that enable the NetScaler appliance to complete non-fully qualified domain names (non-FQDNs) during name resolution. For example, during the process of resolving the domain name abc (which is not fully qualified), if a DNS suffix example.com is configured, the appliance appends the suffix to the domain name (abc.example.com) and resolves it. If DNS suffixes are not configured, the appliance appends a period to the non-FQDNs and resolves the domain name.

Creating DNS Suffixes

DNS suffixes have significance and are valid only when the NetScaler is configured as an end resolver or forwarder. You can specify a suffix of up to 127 characters.

To create DNS suffixes by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a DNS suffix and verify the configuration:

- add dns suffix <dnsSuffix>
- show dns suffix <dnsSuffix>

Example

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)      Suffix: example.com
 Done
>
```

To remove a DNS suffix by using the NetScaler command line, at the NetScaler command prompt, type the rm dns suffix command and the name of the DNS suffix.

To create DNS suffixes by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **DNS Suffix**.
2. In the details pane, click **Add**.
3. In the **Create DNS Suffix** dialog box, type the suffix (for example, **example.com**).
4. Click **Create**, and then click **Close**.

DNS ANY Query

An ANY query is a type of DNS query that retrieves all records available for a domain name. The ANY query must be sent to a name server that is authoritative for a domain.

Behavior in ADNS Mode

In the ADNS mode, the NetScaler appliance returns the records held in its local cache. If there are no records in the cache, the appliance returns the NXDOMAIN (negative) response.

If the NetScaler can match the domain delegation records, it returns the NS records. Otherwise, it returns the NS records of the root domain.

Behavior in DNS Proxy Mode

In proxy mode, the NetScaler appliance checks its local cache. If there are no records in the cache, the appliance passes the query to the server.

Behavior for GSLB Domains

If a GSLB domain is configured on the NetScaler appliance and an ANY query is sent for the GSLB domain (or GSLB site domain), the appliance returns the IP address of the GSLB service that it selects through the Load Balancing decision. If the multiple IP response (MIR) option is enabled, the IP addresses of all GSLB services are sent.

For the NetScaler to return these records when it responds to the ANY query, all records corresponding to a GSLB domain must be configured on the NetScaler.

Note: If records for a domain are distributed between the NetScaler and a server, only records configured on the NetScaler are returned.

The NetScaler provides the option to configure DNS views and DNS policies. These are used for performing global server load balancing. For more information, see Global Server Load Balancing.

Domain Name System Security Extensions

DNS Security Extensions (DNSSEC) is an Internet Engineering Task Force (IETF) standard that aims to provide data integrity and data origin authentication in communications between name servers and clients while still transmitting User Datagram Protocol (UDP) responses in clear text. DNSSEC specifies a mechanism that uses asymmetric key cryptography and a set of new resource records that are specific to its implementation.

The DNSSEC specification is described in RFC 4033, “DNS Security Introduction and Requirements,” RFC 4034, “Resource Records for the DNS Security Extensions,” and RFC 4035, “Protocol Modifications for the DNS Security Extensions.” The operational aspects of implementing DNSSEC within DNS are discussed in RFC 4641, “DNSSEC Operational Practices.”

You can configure DNSSEC on the Citrix® NetScaler® appliance. You can generate and import keys for signing DNS zones. You can configure DNSSEC for zones for which the NetScaler appliance is authoritative. You can configure the NetScaler appliance as a DNS proxy server for signed zones hosted on a farm of back-end name servers. If the NetScaler appliance is authoritative for a subset of the records belonging to a zone for which the appliance is configured as a DNS proxy server, you can include the subset of records in the DNSSEC implementation.

Configuring DNSSEC

Configuring DNSSEC involves enabling DNSSEC on the Citrix® NetScaler® appliance, creating a Zone Signing Key and a Key Signing Key for the zone, adding the two keys to the zone, and then signing the zone with the keys.

Enabling and Disabling DNSSEC

You must enable DNSSEC on the NetScaler appliance for the appliance to respond to DNSSEC-aware clients. By default, DNSSEC is enabled.

You can disable the DNSSEC feature if you do not want the NetScaler appliance to respond to clients with DNSSEC-specific information.

To enable or disable DNSSEC by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable or disable DNSSEC and verify the configuration:

- set dns parameter **-dnssec** (ENABLED | DISABLED)
- show dns parameter

Example

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
       DNS parameters:
       DNS retries: 5
        .
        .
        .
        DNSEC Extension: ENABLED
       Max DNS Pipeline Requests: 255
 Done
>
```

Parameters for enabling and disabling DNSSEC

dnssec

Enable or disable DNSSEC on the NetScaler appliance. Possible values: ENABLED, DISABLED. Default value: ENABLED

To enable or disable DNSSEC by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details pane, click **Change DNS settings**.
3. In the **Configure DNS Parameters** dialog box, select or clear the **Enable DNSSEC Extension** check box.

Creating DNS Keys for a Zone

For each DNS zone that you want to sign, you must create two pairs of asymmetric keys. One pair, called the Zone Signing Key, is used to sign all the resource record sets in the zone. The second pair is called the Key Signing Key and is used to sign only the DNSKEY resource records in the zone.

When the Zone Signing Key and Key Signing Key are created, the suffix .key is automatically appended to the names of the public components of the keys and the suffix .private is automatically appended to the names of their private components.

Additionally, the appliance also creates a Delegation Signer (DS) record and appends the suffix .ds to the name of the record. If the parent zone is a signed zone, you must publish the DS record in the parent zone to establish the chain of trust.

When you create a key, the key is stored in the /nsconfig/dns/ directory, but it is not automatically published in the zone. After you create a key by using the create dns key command, you must explicitly publish the key in the zone by using the add dns key command. The process of generating a key has been separated from the process of publishing the key in a zone to enable you to use alternative means to generate keys. For example, you can import keys generated by other key-generation programs (such as `bind-keygen`) by using Secure File Transfer Protocol (SFTP) and then publish the keys in the zone. For more information about publishing a key in a zone, see Publishing a DNS Key in a Zone.

Perform the steps described in this topic to create a Zone Signing Key and then repeat the steps to create a Key Signing Key. The example that follows the command syntax first creates a Zone Signing Key pair for the zone `example.com`. The example then uses the command to create a Key Signing Key pair for the zone.

To create a DNS key by using the NetScaler command line

At the NetScaler command prompt, type the following command to create a DNS key:

create dns key **-zoneName** <string> **-keyType** <keyType> **-algorithm** RSASHA1 **-keySize** <positive_integer> **-fileNamePrefix** <string>

Example

```
> create dns key -zoneName example.com -keyType zsk -algorithm RSASHA1 -keySize 1024 -fileNamePrefix e
File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /nsconfig/dns/example.com.zsk.rsas
This operation may take some time, Please wait...
 Done
```

```
> create dns key -zoneName example.com -keyType ksk -algorithm RSASHA1 -keySize 4096 -fileNamePrefix e
File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /nsconfig/dns/example.com.ksk.rsas
This operation may take some time, Please wait...
 Done
>
```

Parameters for creating a DNS Key

zoneName

The name of the zone for which the key is being added. This is a mandatory argument.

keyType

The type of key. This is a mandatory argument. Possible values: KSK, KeySigningKey, ZSK, ZoneSigningKey.

Default value: ZSK

algorithm

The algorithm that must be used to generate the keys. This is a mandatory argument. Possible values: RSASHA1. Default value: RSASHA1.

keySize

The key strength. This is a mandatory argument. Default value: 512

fileNamePrefix

A common prefix for the public and private components of the key pair. During key generation, the .key and .private suffixes are appended automatically to the file name prefix.

To create a DNS key by using the NetScaler configuration utility

1. In the navigation pane, click **DNS**.
2. In the details area, click **Create DNS Key**.
3. In the **Create DNS Key** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a DNS Key," as shown:
   - **Zone Name***—zoneName
   - **Type***—keyType
   - **Algorithm***—algorithm
   - **Size***—keySize
   - **File Name Prefix***—fileNamePrefix

   * A required parameter

   Note: For **File Name Prefix**, if you want to modify the file name prefix of an existing key, click the arrow next to the **Browse** button, click either **Local** or **Appliance** (depending on whether the existing key is stored on your local computer or in the /nsconfig/dns/ directory on the appliance), browse to the location of the key, and then double-click the key. The **File Name Prefix** box is populated with only the prefix of the existing key. Modify the prefix accordingly.
4. Click **Create**, and then click **Close**.

Publishing a DNS Key in a Zone

A key (Zone Signing Key or Key Signing Key) is published in a zone by adding the key to the NetScaler appliance. A key must be published in a zone before you sign the zone.

Before you publish a key in a zone, the key must be available in the /nsconfig/dns/ directory. Therefore, if you used other means to generate the key—means other then the create dns key command on the NetScaler appliance (for example, by using the `bind-keygen` program on another computer)—make sure that the key is added to the /nsconfig/dns/ directory before you publish the key in the zone.

If the key has been generated by another program, you can import the key to your local computer and use the NetScaler configuration utility to add the key to the /nsconfig/dns/ directory. Or, you can use other means to import the key to the directory, such as the Secure File Transfer Protocol (SFTP).

You must use the add dns key command for each public-private key pair that you want to publish in a given zone. If you created a Zone Signing Key pair and a Key Signing Key pair for a zone, use the add dns key command to first publish one of the key pairs in the zone and then repeat the command to publish the other key pair. For each key that you publish in a zone, a DNSKEY resource record is created in the zone.

The example that follows the command syntax first publishes the Zone Signing Key pair (that was created for the `example.com` zone) in the zone. The example then uses the command to publish the Key Signing Key pair in the zone.

To publish a key in a zone by using the NetScaler command line

At the NetScaler command prompt, type the following command to publish a key in a zone and verify the configuration:

- add dns key <keyName> <publickey> <privatekey> [**-expires** <positive_integer> [<units>]] [**-notificationPeriod** <positive_integer> [<units>]] [**-TTL** <secs>]
- show dns zone [<zoneName> | **-type** <type>]

Example

```
> add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.com.zsk.rsasha1.1024.private
 Done
> add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.com.ksk.rsasha1.4096.private
 Done
> show dns zone example.com
```

```
        Zone Name : example.com
        Proxy Mode : NO
        Domain Name : example.com
              Record Types : NS SOA DNSKEY
        Domain Name : ns1.example.com
              Record Types : A
        Domain Name : ns2.example.com
              Record Types : A
 Done
>
```

Parameters for publishing a key in a zone

keyName

The name given to a public-private key pair. This is a mandatory argument. Maximum length: 31.

publickey

File name of the public key that is used for signing the zone. This is a mandatory argument. Maximum length: 63

privatekey

File name of the private key that is used for signing the zone. This is a mandatory argument. Maximum length: 63

expires

Time for which the key is valid. Default value: 120 days. Minimum value: 1. Maximum value: 32767.

units

Units for the expiry time. Possible values: MINUTES, HOURS, DAYS. Default value: DAYS.

notificationPeriod

Number of days, hours, or minutes prior to expiry of a key when a notification should be generated. Default value: 7 days. Minimum value: 1. Maximum value: 32767.

units

Units for the notification period. Possible values: MINUTES, HOURS, DAYS. Default value: DAYS.

TTL

Time to Live, in seconds. Default value: 3600. Minimum value: 0. Maximum value: 2147483647.

To publish a key in a DNS zone by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Keys**.
2. In the details pane, click **Add**.
3. In the **Add DNS Key** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for adding a key to the NetScaler appliance," as shown:
   - **DNS Key Name*** —keyName
   - **Public Key*** —publickey
   - **Private Key*** —privatekey
   - **Expires** —expires
   - **Notification Period** —notificationPeriod
   - **TTL** —TTL

   * A required parameter

   Note: For **Public Key** and **Private Key**, to add a key that is stored on your local computer, click the arrow next to the **Browse** button, click **Local**, browse to the location of the key, and then double-click the key.
4. Click **Create**, and then click **Close**.

Configuring a DNS Key

You can configure the parameters of a key that has been published in a zone. You can modify the key's expiry time period, notification period, and time-to-live (TTL) parameters. If you change the expiry time period of a key, the NetScaler appliance automatically re-signs all the resource records in the zone with the key, provided that the zone is currently signed with the particular key.

To configure a key by using the NetScaler command line

At the NetScaler command prompt, type the following command to configure a key and verify the configuration:

- set dns key <keyName> [**-expires** <positive_integer> [<units>]] [**-notificationPeriod** <positive_integer> [<units>]] [**-TTL** <secs>]
- show dns key [<keyName>]

Example

```
> set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3 DAYS -TTL 3600
 Done
> show dns key example.com.ksk
1)      Key Name: example.com.ksk
        Expires: 30 DAYS        Notification: 3 DAYS    TTL: 3600
        Public Key File: example.com.ksk.rsasha1.4096.key
        Private Key File: example.com.ksk.rsasha1.4096.private
 Done
>
```

Parameters for configuring a key

keyName

The name given to a public/private key pair. This is a mandatory argument. Maximum length: 31.

expires

Time for which the key is valid. Default value: 120 days. Minimum value: 1. Maximum value: 32767.

units

Units for the expiry time. Possible values: MINUTES, HOURS, DAYS. Default value: DAYS.

notificationPeriod

Number of days, hours, or minutes prior to expiry of a key when a notification should be generated. Default value: 7 days. Minimum value: 1. Maximum value: 32767.

units

Units for the notification period Possible values: MINUTES, HOURS, DAYS. Default value: DAYS.

TTL

Time to Live, in seconds. Default value: 3600. Minimum value: 0. Maximum value: 2147483647.

To configure a key by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Keys**.
2. In the details pane, click the key that you want to configure, and then click **Open**.
3. In the **Configure DNS Key** dialog box, modify the values of the following parameters, which correspond to parameters described in "Parameters for configuring a key," as shown:
   - **Expires**—expires
   - **Notification Period**—notificationPeriod
   - **TTL**—TTL
4. Click **OK**.

Signing and Unsigning a DNS Zone

To secure a DNS zone, you must sign the zone with the keys that have been published in the zone. When you sign a zone, the NetScaler appliance creates a Next Secure (NSEC) resource record for each owner name. Then, it uses the Key Signing Key to sign the DNSKEY resource record set. Finally, it uses the Zone Signing Key to sign all the resource record sets in the zone, including the DNSKEY resource record sets and NSEC resource record sets. Each sign operation results in a signature for the resource record sets in the zone. The signature is captured in a new resource record called the RRSIG resource record.

After you sign a zone, you must save the configuration.

To sign a zone by using the NetScaler command line

At the NetScaler command prompt, type the following command to sign a zone and verify the configuration:

- sign dns zone <zoneName> [**-keyName** <string> ...]
- show dns zone [<zoneName> | **-type** (ADNS | PROXY | ALL)]
- save config

Example

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
        Zone Name : example.com
        Proxy Mode : NO
        Domain Name : example.com
                Record Types : NS SOA DNSKEY RRSIG NSEC
        Domain Name : ns1.example.com
                Record Types : A RRSIG NSEC
        Domain Name : ns2.example.com
                Record Types : A RRSIG
        Domain Name : ns2.example.com
                Record Types : RRSIG NSEC
Done
> save config
 Done
>
save config
```

To unsign a zone by using the NetScaler command line

At the NetScaler command prompt, type the following command to unsign a zone and verify the configuration:

- unsign dns zone <zoneName> [**-keyName** <string> ...]
- show dns zone [<zoneName> | **-type** (ADNS | PROXY | ALL)]

Example

```
> unsign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
        Zone Name : example.com
        Proxy Mode : NO
        Domain Name : example.com
                Record Types : NS SOA DNSKEY
        Domain Name : ns1.example.com
                Record Types : A
        Domain Name : ns2.example.com
                Record Types : A
 Done
>
```

Parameters for signing and unsigning a DNS zone

zoneName

The name of the zone being signed. This is a mandatory argument. Maximum length: 255.

keyName

The name given to a public/private key pair. Maximum length: 127

To sign or unsign a zone by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Zones**.
2. In the details pane, click the zone that you want to sign, and then click **Sign/Unsign**.
3. In the **Sign/Unsign DNS Zone** dialog box, do one of the following:
   - To sign the zone, select the check boxes for the keys (Zone Signing Key and Key Signing Key) with which you want to sign the zone.

     You can sign the zone with more than one Zone Signing Key or Key Signing Key pair.
   - To unsign the zone, clear the check boxes for the keys (Zone Signing Key and Key Signing Key) with which you want to unsign the zone.

     You can unsign the zone with more than one Zone Signing Key or Key Signing Key pair.
4. Click **OK**.

Viewing the NSEC Records for a Given Record in a Zone

You can view the NSEC records that the NetScaler appliance automatically creates for each owner name in the zone.

To view the NSEC record for a given record in a zone by using the NetScaler command line

At the NetScaler command prompt, type the following command to view the NSEC record for a given record in a zone:

show dns nsecRec [<hostName> | **-type** (ADNS | PROXY | ALL)]

Example

```
> show dns nsecRec example.com
1)      Domain Name : example.com
       Next Nsec Name: ns1.example.com
       Record Types : NS SOA DNSKEY RRSIG NSEC
 Done
>
```

Parameters for viewing the NSEC record for a given record in a zone

hostName

The domain name whose information is to be displayed.

Maximum length: 255

type

The NSEC record type. The type can take 3 values:

ADNS - If this is specified, all of the authoritative NSEC records will be displayed.

PROXY - If this is specified, all of the proxy NSEC records will be displayed.

ALL - If this is specified, all of the NSEC records will be displayed.

Possible values: ALL, ADNS, PROXY

To view the NSEC record for given record in a zone by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, expand **Records**, and then click **Next Secure Records**.
2. In the details pane, click the name of the record for which you want to view the NSEC record. The NSEC record for the record you select is displayed in the **Details** area.

Removing a DNS Key

You remove a key from the zone in which it is published when the key has expired or if the key has been compromised. When you remove a key from the zone, the zone is automatically unsigned with the key. Removing the key with this command does not remove the key files present in the /nsconfig/dns/ directory. If the key files are no longer needed, they have to be explicitly removed from the directory.

To remove a key from the NetScaler appliance by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a key and verify the configuration:

- rm dns key <keyName>
- show dns key <keyName>

Example

```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]
>
```

Parameters for removing a key from the NetScaler appliance

keyName

The name given to a public/private key pair. This is a mandatory argument. Maximum length: 31

To remove a key from the NetScaler appliance by using the NetScaler configuration utility

1. In the navigation pane, expand **DNS**, and then click **Keys**.
2. In the details pane, click the name of the key that you want to remove from the appliance, and then click **Remove**.

Configuring DNSSEC When the NetScaler Appliance Is Authoritative for a Zone

When the Citrix® NetScaler® appliance is authoritative for a given zone, all the resource records in the zone are configured on the NetScaler appliance. To sign the authoritative zone, you must create keys (the Zone Signing Key and the Key Signing Key) for the zone, add the keys to the appliance, and then sign the zone, as described in Creating DNS Keys for a Zone, Publishing a DNS Key in a Zone, and Signing and Unsigning a DNS Zone, respectively.

If any global server load balancing (GSLB) domains configured on the appliance belong to the zone being signed, the GSLB domain names are signed along with the other records that belong to the zone.

After you sign a zone, responses to requests from DNSSEC-aware clients include the RRSIG resource records along with the requested resource records. DNSSEC must be enabled on the appliance. For more information about enabling DNSSEC, see Enabling and Disabling DNSSEC.

Finally, after you configure DNSSEC for the authoritative zone, you must save the NetScaler configuration.

Configuring DNSSEC for a Zone for Which the NetScaler Appliance Is a DNS Proxy Server

The procedure for signing a zone for which the Citrix® NetScaler® appliance is configured as a DNS proxy server depends on whether or not all zone information is owned by the back-end name servers. If the back-end name servers own all zone information, the NetScaler configuration for managing the back-end servers is considered a *zone-less DNS proxy server configuration*. However, in some cases, the NetScaler appliance might own a subset of the records that belong to a zone that is owned by the back-end name servers. Such a configuration is considered as a *partial zone ownership configuration*. The basic DNSSEC configuration tasks for both NetScaler configurations are the same. However, signing the partial zone on the NetScaler appliance requires some additional configuration steps to be performed.

Note: The terms *zone-less proxy server configuration* and *partial zone* are used only in the context of the NetScaler appliance.

Configuring DNSSEC for a Zone-Less DNS Proxy Server Configuration

For a zone-less DNS proxy server configuration, zone signing must be performed on the back-end name servers. On the NetScaler appliance, you configure the appliance as a DNS proxy server for the zone. You create a load balancing virtual server of protocol type DNS, configure services on the appliance to represent the name servers, and then bind the services to the load balancing virtual server. For more information about these configuration tasks, see Configuring the NetScaler as a DNS Proxy Server.

When a client sends the appliance a DNS request with the DNSSEC OK (DO) bit set, the appliance checks its cache for the requested information. If the resource records are not available in its cache, the appliance forwards the request to one of the DNS name servers, and then relays the response from the name server to the client. Additionally, the appliance caches the RRSIG resource records along with the response from the name server. Subsequent requests from DNSSEC-aware clients are served from the cache (including the RRSIG resource records), subject to the time-to-live (TTL) parameter. If a client sends a DNS request without setting the DO bit, the appliance responds with only the requested resource records, and does not include the RRSIG resource records that are specific to DNSSEC.

Configuring DNSSEC for a Partial Zone Ownership Configuration

In some NetScaler configurations, even though the authority for a zone lies with the back-end name servers, a subset of the resource records that belong to the zone might be configured on the NetScaler appliance. The NetScaler appliance owns (or is authoritative for) only this subset of records. Such a subset of records can be considered to constitute a *partial zone* on the appliance. The appliance owns the partial zone. All other records are owned by the back-end name servers.

A typical partial zone configuration on the NetScaler appliance is seen when global server load balancing (GSLB) domains are configured on the appliance, and the GSLB domains are a part of a zone for which the back-end name servers are authoritative.

Signing a zone that includes only a partial zone on the NetScaler appliance involves including the partial zone information in the back-end name server zone files, signing the zone on the back-end name servers, and then signing the partial zone on the NetScaler appliance. The same key set must be used to sign the zone on the name servers and the partial zone on the NetScaler appliance.

To sign the zone on the back-end name servers

1. Include the resource records that are contained in the partial zone, in the zone files of the name servers.

2. Create keys and use the keys to sign the zone on the back-end name servers.

To sign the partial zone on the NetScaler appliance

1. Create a zone with the name of the zone that is owned by the back-end name servers. When configuring the partial zone, set the **proxyMode** parameter to YES. This zone is the partial zone that contains the resource records owned by the NetScaler appliance.

   For example, if the name of the zone that is configured on the back-end name servers is example.com, you must create a zone named example.com on the NetScaler appliance, with the **proxyMode** parameter set to YES. For more information about adding a zone, see Configuring a DNS Zone.

   Note: Do not add SOA and NS records for the zone. These records should not exist on the appliance for a zone for which the appliance is not authoritative.

2. Import the keys (from one of the back-end name servers) to the NetScaler appliance and then add them to the /nsconfig/dns/ directory. For more information about how you can import a key and add it to the NetScaler appliance, see Publishing a DNS Key in a Zone.

3. Sign the partial zone with the imported keys. When you sign the partial zone with the keys, the appliance generates RRSIG and NSEC records for the resource record sets and individual resource records in the partial zone, respectively. For more information about signing a zone, see Signing and Unsigning a DNS Zone.

Configuring DNSSEC for GSLB Domain Names

If global server load balancing (GSLB) is configured on the Citrix® NetScaler® appliance and the appliance is authoritative for the zone to which the GSLB domain names belong, all GLSB domain names are signed when the zone is signed. For more information about signing a zone for which the NetScaler appliance is authoritative, see Configuring DNSSEC When the NetScaler Appliance Is Authoritative for a Zone.

If the GSLB domains belong to a zone for which the back-end name servers are authoritative, you must first sign the zone on the name servers, and then sign the partial zone on the NetScaler appliance to complete the DNSSEC configuration for the zone. For more information, see Configuring DNSSEC for a Partial Zone Ownership Configuration.

Zone Maintenance

From a DNSSEC perspective, zone maintenance involves rolling over Zone Signing Keys and Key Signing Keys when key expiry is imminent. These zone maintenance tasks have to be performed manually. The process of re-signing a zone is performed automatically and does not require manual intervention.

Re-Signing an Updated Zone

When a zone is updated, that is, when new records are added to the zone or existing records are changed, the process of re-signing the new (or modified) record is performed automatically by the Citrix® NetScaler® appliance. If a zone contains multiple Zone Signing Keys, the NetScaler appliance re-signs the new (or modified) record with the key with which the zone is signed at the point in time when the re-signing is to be performed.

Rolling Over DNSSEC Keys

On the NetScaler appliance, you can use the pre-publish and double signature methods to perform a rollover of the Zone Signing Key and Key Signing Key. More information about these two rollover methods is available in RFC 4641, "DNSSEC Operational Practices."

The following topics map commands on the NetScaler appliance to the steps in the rollover procedures discussed in RFC 4641.

The key expiry notification is sent through an SNMP trap called dnskeyExpiry. Three MIB variables, dnskeyName, dnskeyTimeToExpire, and dnskeyUnitsOfExpiry are sent along with the dnskeyExpiry SNMP trap. For more information, see *Citrix NetScaler SNMP OID Reference* at http://support.citrix.com/article/CTX128676.

Pre-Publish Key Rollover

RFC 4641, "DNSSEC Operational Practices" defines four stages for the pre-publish key rollover method: initial, new DNSKEY, new RRSIGs, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the NetScaler appliance. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both Key Signing Keys and Zone Signing Keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.

  Example

  Consider the key, `example.com.zsk1`, with which the zone `example.com` is currently signed. The zone contains only those RRSIGs that were generated by the `example.com.zsk1` key, which is due for expiry. The Key Signing Key is `example.com.ksk1`.

- **Stage 2: New DNSKEY.** A new key is created and published in the zone (that is, the key is added to the NetScaler appliance), but the zone is not signed with the new key until the pre-roll phase is complete. In this stage, the zone contains the old key, the new key, and the RRSIGs generated by the old key. Publishing the new key for the complete duration of the pre-roll phase gives the DNSKEY resource record (that corresponds to the new key) enough time to propagate to the secondary name servers.

  Example

  A new key `example.com.zsk2` is added to the `example.com` zone. The zone is not signed with `example.com.zsk2` until the pre-roll phase is complete. The example.com zone contains DNSKEY resource records for both `example.com.zsk1` and `example.com.zsk2`.

  NetScaler commands

Perform the following tasks on the NetScaler appliance:

- Create a new DNS key by using the create dns key command.

  For more information about creating a DNS key, including an example, see Creating DNS Keys for a Zone.

- Publish the new DNS key in the zone by using the add dns key command.

  For more information about publishing the key in the zone, including an example, see Publishing a DNS Key in a Zone.

- **Stage 3: New RRSIGs.** The zone is signed with the new DNS key and then unsigned with the old DNS key. The old DNS key is not removed from the zone and remains published until the RRSIGs that were generated by the old key expire.

  Example

  The zone is signed with `example.com.zsk2` and then unsigned with `example.com.zsk1`. The zone continues to publish `example.com.zsk1` until the RRSIGs that were generated by `example.com.zsk1` expire.

  NetScaler commands

  Perform the following tasks on the NetScaler appliance:

  - Sign the zone with the new DNS key by using the sign dns zone command.
  - Unsign the zone with the old DNS key by using the unsign dns zone command.

  For more information about signing and unsigning a zone, including examples, see Signing and Unsigning a DNS Zone.

- **Stage 4: DNSKEY Removal.** When the RRSIGs that were generated by the old DNS key expire, the old DNS key is removed from the zone.

  Example

  The old DNS key `example.com.zsk1` is removed from the `example.com` zone.

  NetScaler commands

  On the NetScaler appliance, you remove the old DNS key by using the rm dns key command. For more information about removing a key from a zone, including an example, see Removing a DNS Key.

Double Signature Key Rollover

RFC 4641, “DNSSEC Operational Practices” defines three stages for double signature key rollover: initial, new DNSKEY, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the NetScaler appliance. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both Key Signing Keys and Zone Signing Keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.

  Example

  Consider the key, `example.com.zsk1`, with which the zone `example.com` is currently signed. The zone contains only those RRSIGs that were generated by the `example.com.zsk1` key, which is due for expiry. The Key Signing Key is `example.com.ksk1`.

- **Stage 2: New DNSKEY.** The new key is published in the zone and the zone is signed with the new key. The zone contains the RRSIGs that are generated by the old and the new keys. The minimum duration for which the zone must contain both sets of RRSIGs is the time required for all the RRSIGs to expire.

  Example

  A new key `example.com.zsk2` is added to the `example.com` zone. The zone is signed with `example.com.zsk2`. The `example.com` zone now contains the RRSIGs generated from both keys.

  NetScaler commands

  Perform the following tasks on the NetScaler appliance:

  - Create a new DNS key by using the create dns key command.

    For more information about creating a DNS key, including an example, see Creating DNS Keys for a Zone.

  - Publish the new key in the zone by using the add dns key command.

    For more information about publishing the key in the zone, including an example, see Publishing a DNS Key in a Zone.

  - Sign the zone with the new key by using the sign dns zone command.

    For more information about signing a zone, including examples, see Signing and Unsigning a DNS Zone.

- **Stage 3: DNSKEY Removal.** When the RRSIGs that were generated by the old DNS key expire, the old DNS key is removed from the zone.

  Example

  The old DNS key `example.com.zsk1` is removed from the `example.com` zone.

  NetScaler commands

  On the NetScaler appliance, you remove the old DNS key by using the rm dns key command.

  For more information about removing a key from a zone, including an example, see Removing a DNS Key.

Firewall Load Balancing

Firewall load balancing distributes traffic across multiple firewalls, providing fault tolerance and increased throughput. Firewall load balancing protects your network by:

- Dividing the load between the firewalls, which eliminates a single point of failure and allows the network to scale.
- Increasing high availability.

Configuring a NetScaler appliance for firewall load balancing is similar to configuring load balancing, with the exception that the recommended service type is ANY, recommended monitor type is PING, and the load balancing virtual server mode is set to MAC.

You can set up firewall load balancing in either a sandwich or an enterprise configuration. You configure a firewall load balancing sandwich environment for load balancing traffic entering the network from outside and traffic leaving the network to the internet. You configure two NetScaler appliances, one on each side of a set of firewalls. You configure an enterprise environment for load balancing traffic leaving the network to the internet. You configure a single appliance between the internal network and the firewalls that provide access to the Internet.

Important: If you configure static routes on the NetScaler for the destination IP address and enable L3 mode, the NetScaler uses its routing table to route the traffic instead of sending the traffic to the load balancing vserver.

Note: For FTP to work, an additional virtual server or service should be configured on the NetScaler with IP address and port as * and 21 respectively, and the service type specified as FTP. In this case, the NetScaler manages the FTP protocol by accepting the FTP control connection, modifying the payload, and managing the data connection, all through the same firewall.

Firewall Load Balancing supports only some of the load balancing methods supported on the NetScaler. Also, you can configure only a few types of persistence and monitors.

Firewall Load Balancing Methods

The following load balancing methods are supported for firewall load balancing.

- Least Connections
- Round Robin
- Least Packets
- Least Bandwidth
- Source IP Hash

- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port hash
- Least Response Time Method (LRTM)
- Custom Load

For more information about load balancing methods (algorithms), see Load Balancing Algorithms.

Firewall Persistence

Only SOURCEIP, DESTIP, and SOURCEIPDESTIP based persistence are supported for firewall load balancing.

For more information about configuring persistence, see Persistence and Persistent Connections.

Firewall Server Monitoring

Only PING and transparent monitors are supported in firewall load balancing. You can bind a PING monitor (default) to the backend service that represents the firewall. If a firewall is configured not to respond to ping packets, you can configure transparent monitors to monitor hosts on the trusted side through individual firewalls.

For more information on configuring monitors, see Monitors.

Sandwich Environment

A NetScaler deployment in a sandwich mode is capable of load balancing network traffic through firewalls in both directions: ingress (traffic entering the network from the outside, such as the internet) and egress (traffic leaving the network to the internet).

In this setup, a NetScaler is located on each side of a set of firewalls. The NetScaler placed between the firewalls and the Internet, called the *external* NetScaler that handles ingress traffic selects the best firewall, based on the configured method. The NetScaler between the firewalls and the private network, called the *internal* NetScaler tracks the firewall from which the initial packet for a session is received. It then makes sure that all subsequent packets for that session are sent to the same firewall.

The internal NetScaler can be configured as a regular traffic manager to load balance traffic across the private network servers. This configuration also allows traffic originating from the private network (egress) to be load balanced across the firewalls.

The following diagram shows the sandwich firewall load balancing environment.

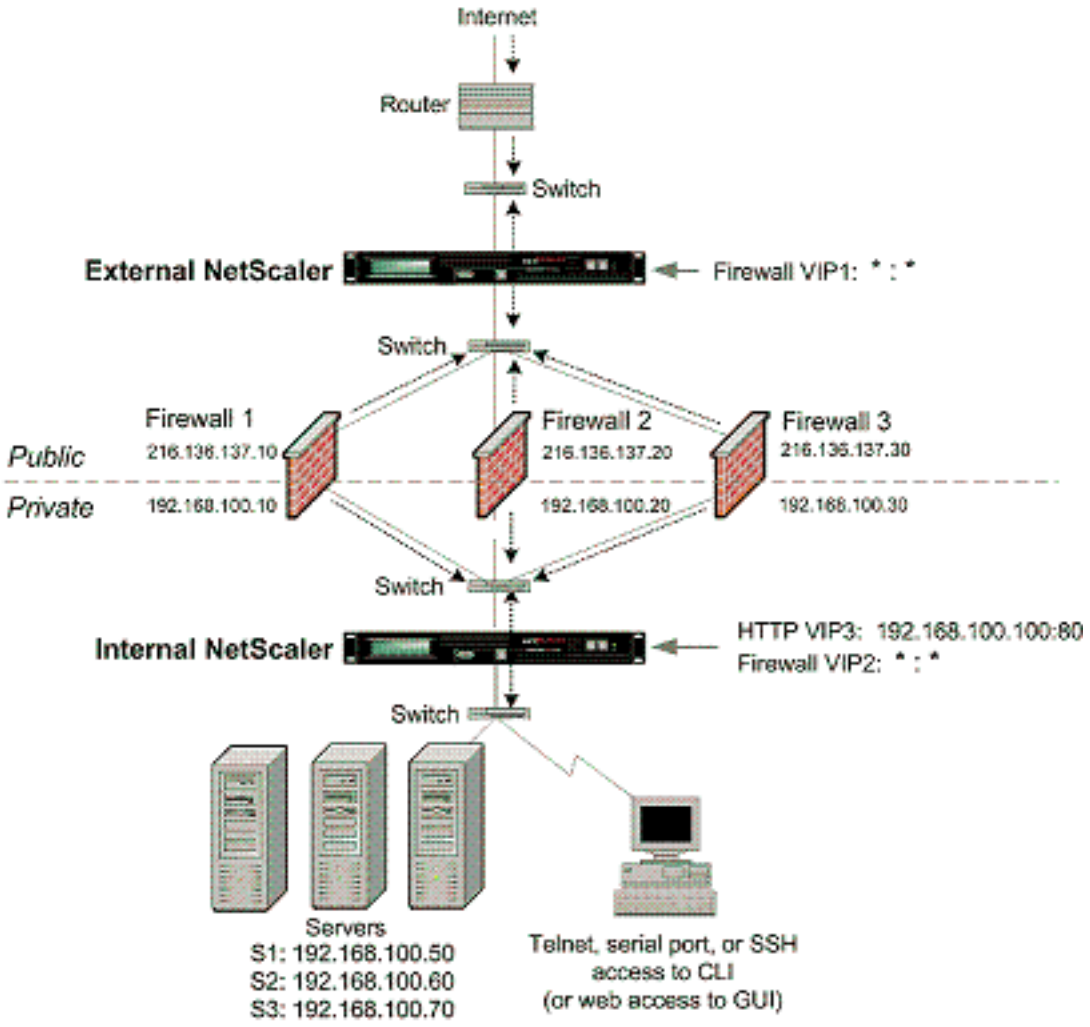


Figure 1. Firewall Load Balancing (Sandwich)

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Sandwich Environment

A NetScaler deployment in a sandwich mode is capable of load balancing network traffic through firewalls in both directions: ingress (traffic entering the network from the outside, such as the internet) and egress (traffic leaving the network to the internet).

In this setup, a NetScaler is located on each side of a set of firewalls. The NetScaler placed between the firewalls and the Internet, called the *external* NetScaler that handles ingress traffic selects the best firewall, based on the configured method. The NetScaler between the firewalls and the private network, called the *internal* NetScaler tracks the firewall from which the initial packet for a session is received. It then makes sure that all subsequent packets for that session are sent to the same firewall.

The internal NetScaler can be configured as a regular traffic manager to load balance traffic across the private network servers. This configuration also allows traffic originating from the private network (egress) to be load balanced across the firewalls.

The following diagram shows the sandwich firewall load balancing environment.

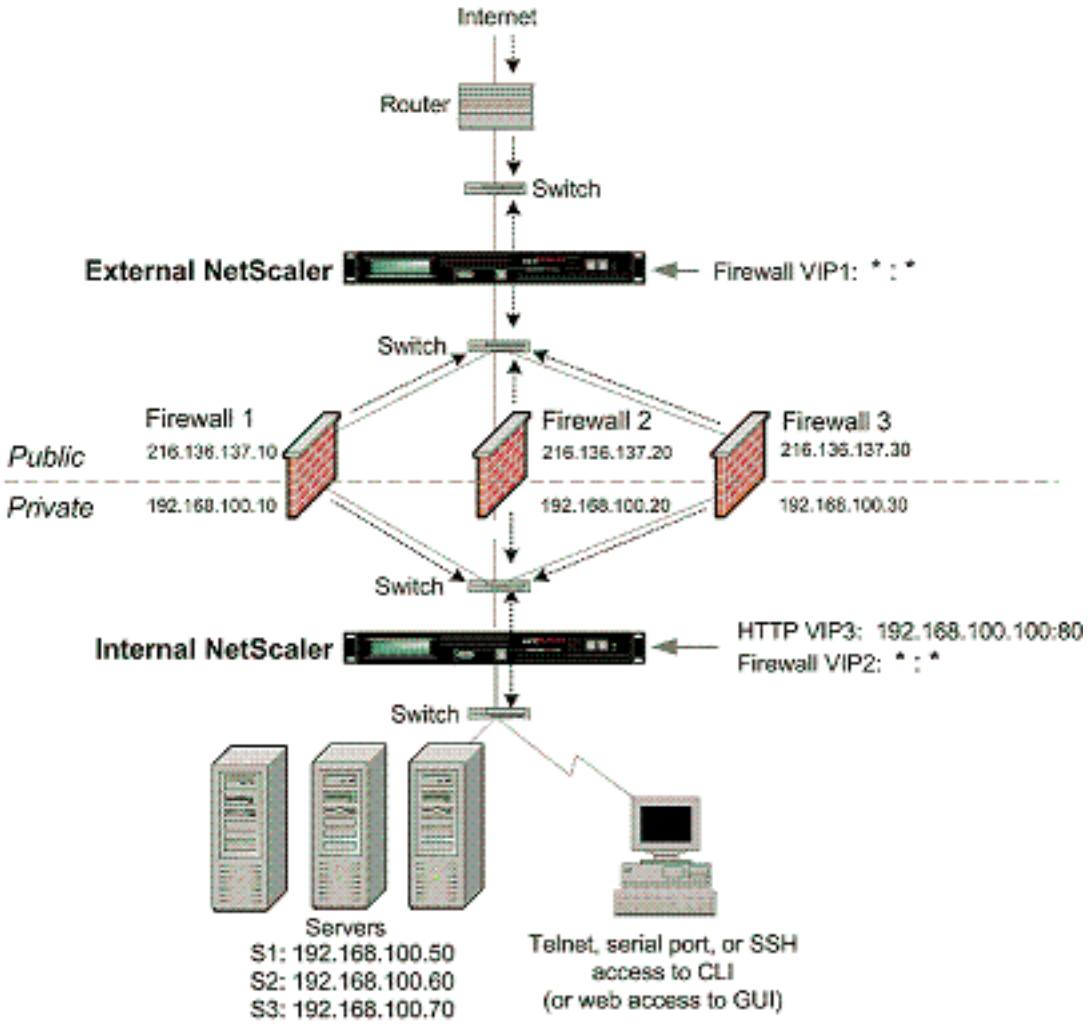


Figure 1. Firewall Load Balancing (Sandwich)

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the External NetScaler in a Sandwich Environment

Perform the following tasks to configure the external NetScaler in a sandwich environment

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server for traffic coming from the Internet.
- Configure the virtual server in MAC rewrite mode.
- Bind services to the wildcard virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

To enable load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following command to enable load balancing and verify the configuration:

- enable feature lb
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging               WL              OFF
```

```
2)    Surge Protection          SP              ON
3)    Load Balancing            LB              ON
.
.
.
24)   NetScaler Push            push            OFF
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Load Balancing** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the NetScaler command line

At the NetScaler command prompt, type:

add service <name> <serverName> ANY *

Example

add service Service-HTTP-1 10.102.29.5 ANY *

Parameters for configuring a wildcard service for each firewall

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP. For a wildcardservice, specify a service type of ANY.

port

Port on which the service listens. The port number must be a positive number not greater than 65535. For a wildcard service, specify an asterisk (*) as the port number.

Note: For more information about the SSL and SSL_TCP service types, see Secure Sockets Layer (SSL) Acceleration.

To configure a wildcard service for each firewall by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a wildcard service for each firewall” as shown:
   - **Service Name**—name
   - **Server**—serverName
4. In **Protocol**, select **ANY**, and in **Port**, select ***.
5. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the

destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [**-destIP** <ip_addr|ipv6_addr|*>] [**-transparent** (**YES** | **NO**)]
- bind lb monitor <monitorName> <serviceName>

Example

add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
To bind a PING monitor, type the following command:
bind monitor PING fw-svc1

Parameters for configuring a monitor

monitorName (Name)

The name of the monitor. This is a mandatory argument. Maximum Length: 31.

type (Type)

The type of monitor. This is a mandatory argument. Default: PING.

destIP (Destination IP)

The IP address to which the probe is sent. If the destination IP address is set to 0, the destination IP address is that of the server to which the monitor is bound. Default value: 0

transparent (Transparent)

The state of the monitor for transparent devices, such as firewalls, based on the responsiveness of the services behind them. If the monitoring of transparent devices is enabled, the destination IP address should be specified. The probe is sent to the specified destination IP address using the MAC address of the transparent device. Possible values: YES, NO. Default value: NO

To create and bind a transparent monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor dialog** box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a monitor" as shown:
   - **Name***
   - **Type***
   - **Destination IP**
   - **Transparent**

   * A required parameter
4. Click **Create**, and then click **Close**. In the **Monitors** pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure a wildcard virtual server for traffic coming from the Internet

To configure a wildcard virtual server for traffic coming from the Internet by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> ANY * *

Example

add lb vserver Vserver-LB-1 ANY * *

Parameters for configuring a wildcard virtual server for traffic coming from the Internet

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a wildcard virtual server for traffic coming from the Internet” as shown:
   - **Name**—name
4. In **Protocol**, select **ANY**, and in **IP Address** and **Port**, select ***.
5. Click **Create**, and then click **Close**. The virtual server you created appears in the **Load Balancing Virtual Servers** pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**m** <RedirectionMode>

Example

set lb vserver Vserver-LB-1 -m MAC

Parameter for configuring the virtual server in MAC rewrite mode

m

The load balancing redirection mode. Possible Values: IP, MAC. Default: IP.

If set to IP, the destination IP address of the request is changed to the IP address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server.

If set to MAC, the destination MAC address is changed to the MAC address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server. With this setting, the destination IP address of the traffic is not changed.

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, under **Redirection Mode**, click **MAC-Based**.
4. Click **OK**.

Bind services to the wildcard virtual server

To bind a service to the wildcard virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <name> <serviceName>

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind a service to the wildcard virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service (for example, **Vserver-LB-1**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Service-HTTP-1**).
5. Click **OK**.

Note: You can bind a service to multiple virtual servers.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save config
- show vserver <name>

Example

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (*:*) - ANY    Type: ADDRESS
```

```
        State: UP
        Last state change was at Mon Jun 14 06:40:14 2010
        Time since last state change: 0 days, 00:00:11.240
        Effective State: UP  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: SRCIPDESTIPHASH
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: *) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
        fw-svc1 (10.102.29.251:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 10:04:50 2010
        Time since last state change: 0 days, 00:00:38.120
        Server Name: 10.102.29.251
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 5       Failed [Total: 5 Current: 5]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.415 millisec
 Done
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click **Save**.
2. In the **Save Config** dialog box, click **Yes**.
3. In the navigation pane, click **Load Balancing**, and then click **Virtual Servers**.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the **Details** pane are correct.
5. In the navigation pane, click **Load Balancing**, and then click **Services**.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the **Details** pane are correct.

Configuring the Internal NetScaler in a Sandwich Environment

Perform the following tasks to configure the internal NetScaler in a sandwich environment

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following command to enable load balancing and verify the configuration:

- enable feature lb
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                 Acronym           Status
      -------                 -------           ------
1)    Web Logging               WL              OFF
2)    Surge Protection          SP              ON
3)    Load Balancing            LB              ON
.
.
.
24)   NetScaler Push            push            OFF
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Load Balancing** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the NetScaler command line

At the NetScaler command prompt, type:

add service <name> <serverName> ANY *

Example

```
add service Service-HTTP-1 10.102.29.5 ANY *
```

Parameters for configuring a wildcard service for each firewall

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP. For a wildcardservice, specify a service type of ANY.

port

Port on which the service listens. The port number must be a positive number not greater than 65535. For a wildcard service, specify an asterisk (*) as the port number.

Note: For more information about the SSL and SSL_TCP service types, see Secure Sockets Layer (SSL) Acceleration.

To configure a wildcard service for each firewall by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a wildcard service for each firewall" as shown:
   - **Service Name**—name
   - **Server**—serverName
4. In **Protocol**, select **ANY**, and in **Port**, select ***.
5. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [**-destIP** <ip_addr|ipv6_addr|*>] [**-transparent** (**YES** | **NO**)]
- bind lb monitor <monitorName> <serviceName>

Example

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

Parameters for configuring a monitor

monitorName (Name)

The name of the monitor. This is a mandatory argument. Maximum Length: 31.

type (Type)

The type of monitor. This is a mandatory argument. Default: PING.

destIP (Destination IP)

The IP address to which the probe is sent. If the destination IP address is set to 0, the destination IP address is that of the server to which the monitor is bound. Default value: 0

transparent (Transparent)

The state of the monitor for transparent devices, such as firewalls, based on the responsiveness of the services behind them. If the monitoring of transparent devices is enabled, the destination IP address should be specified. The probe is sent to the specified destination IP address using the MAC address of the transparent device. Possible values: YES, NO. Default value: NO

To create and bind a transparent monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a monitor", as shown:
   - **Name***
   - **Type***
   - **Destination IP**
   - **Transparent**

   * A required parameter
4. Click **Create**, and then click **Close**. In the **Monitors** pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> ANY * *

Example

add lb vserver Vserver-LB-1 ANY * *

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a virtual server” as shown:
   - Name—name
4. In **Protocol**, select **ANY**, and in **IP Address** and **Port**, select ***.
5. Click **Create**, and then click **Close**. The virtual server you created appears in the **Load Balancing Virtual Servers** pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**m** <RedirectionMode>

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

Parameter for configuring the virtual server in MAC rewrite mode

m

The load balancing redirection mode. Possible Values: IP, MAC. Default: IP.

If set to IP, the destination IP address of the request is changed to the IP address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server.

If set to MAC, the destination MAC address is changed to the MAC address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server. With this setting, the destination IP address of the traffic is not changed.

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, under **Redirection Mode**, click **MAC-Based**.
4. Click **OK**.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

```
bind lb vserver <name> <serviceName>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service (for example, **Vserver-LB-1**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Service-HTTP-1**).
5. Click **OK**.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

To configure a service for each virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

add service <name> <serverName> HTTP <port>

Example

add service Service-HTTP-1 10.102.29.5 HTTP 80

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously-created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format.

When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

Note: For more information about the SSL and SSL_TCP service types, see Secure Sockets Layer (SSL) Acceleration.

To configure a service for each virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service” as shown:
   - **Service Name**—name
   - **Server**—serverName
   - **Port**—port
4. In **Protocol**, specify **HTTP**. Under **Available Monitors**, select **HTTP**.
5. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Configure a monitor for each service

To bind a monitor to a service by using the NetScaler command line

At the NetScaler command prompt, type:

bind mon <MonitorName> <ServiceName>

Example

bind mon monitor-HTTP-1 Service-HTTP-1

To bind a monitor to a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to bind the monitor (for example, **Service-HTTP-1**), and then click **Open**.
3. On the **Monitors** tab, in the **Available** list box, select the monitor you want to bind the service (for example, monitor-HTTP-1), and then click **Add**.
4. In the **Configured** box, click **OK**.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> HTTP <ip> <port>

Example

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a virtual server” as shown:
   - Name—name
   - IP Address—IPAddress

     Note: If the virtual server uses IPv6, select the **IPv6** check box and enter the address in IPv6 format (for example, **1000:0000:0000:0000:0005:0600:700a:888b**).
   - Port—port
4. Under **Protocol**, select **HTTP**.
5. Click **Create**, and then click **Close**. The virtual server you created appears in the **Load Balancing Virtual Servers** pane.

Bind HTTP services to the HTTP virtual server

To bind HTTP services to the wildcard virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <name> <serviceName>

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind HTTP services to the wildcard virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service (for example, **Vserver-LB-1**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Service-HTTP-1**).
5. Click **OK**.

Note: You can bind a service to multiple virtual servers.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save config
- show vserver <name>

Example

```
save config
show lb vserver FWLBVIP2
      FWLBVIP2 (*:*) - ANY    Type: ADDRESS
```

```
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click **Save**.
2. In the **Save Config** dialog box, click **Yes**.
3. In the navigation pane, click **Load Balancing**, and then click **Virtual Servers**.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the **Details** pane are correct.
5. In the navigation pane, click **Load Balancing**, and then click **Services**.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the **Details** pane are correct.

Monitoring a Firewall Load Balancing Setup in a Sandwich Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the NetScaler command line

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the NetScaler command prompt, type:

stat lb vserver [-**detail**] **[<name>]**

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
```

```
                  vsvrIP  port     Protocol       State    Req/s   Hits/s
One                  *    80         HTTP          UP      5/s      0/s
Two                  *     0          TCP        DOWN      0/s      0/s
Three                *  2598          TCP        DOWN      0/s      0/s
dnsVirtualNS    10.102.29.90    53          DNS        DOWN      0/s      0/s
BRVSERV          10.10.1.1    80         HTTP        DOWN      0/s      0/s
LBVIP          10.102.29.66    80         HTTP          UP      0/s      0/s
 Done
```

Parameters for displaying statistics

detail

Include the statistics for hits per second and the total number of hits.

name

Name of the virtual server whose statistics are displayed.

To display virtual server statistics by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.
3. In the details pane, click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the NetScaler command line

At the NetScaler command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service whose statistics you want to view (for example, **Service-HTTP-1**).
3. Click **Statistics**. The statistics appear in a new window.

Enterprise Environment

In the enterprise setup, the NetScaler is placed between the firewalls connecting to the public Internet and the internal private network and handles egress traffic. The NetScaler selects the best firewall based on the configured load balancing policy.

The following diagram shows the enterprise firewall load balancing environment

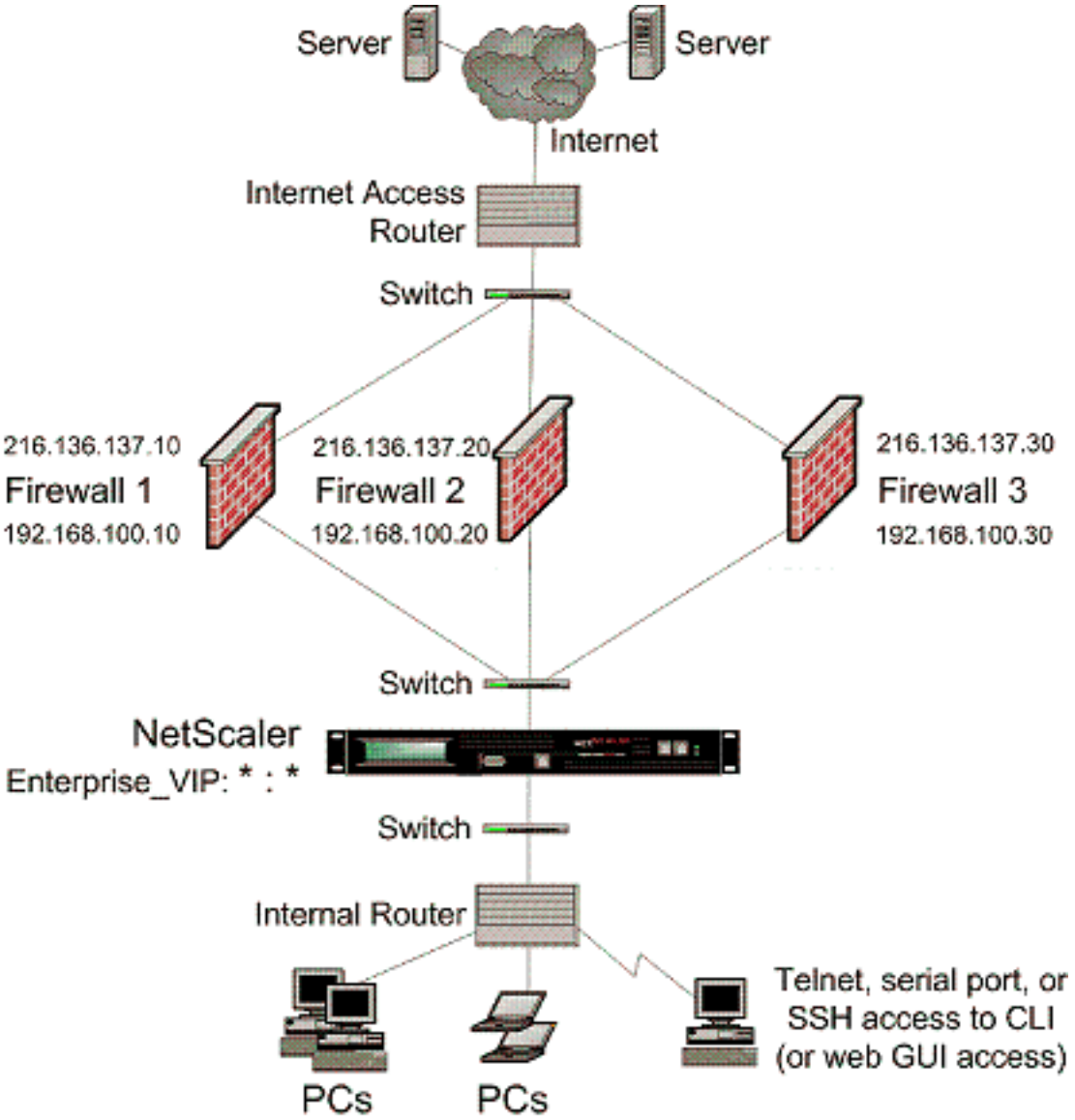


Figure 1. Firewall Load Balancing (Enterprise)

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and vserver with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the NetScaler in an Enterprise Environment

Perform the following tasks to configure a NetScaler in an enterprise environment.

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following command to enable load balancing and verify the configuration:

- enable feature lb
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

       Feature                 Acronym           Status
       -------                 -------           ------
1)     Web Logging               WL               OFF
2)     Surge Protection          SP               ON
3)     Load Balancing            LB               ON
.
.
.
24)    NetScaler Push            push             OFF
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Load Balancing** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the NetScaler command line

At the NetScaler command prompt, type:

add service <name> <serverName> ANY *

Example

```
add service Service-HTTP-1 10.102.29.5 ANY *
```

Parameters for configuring a wildcard service for each firewall

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP. For a wildcardservice, specify a service type of ANY.

port

Port on which the service listens. The port number must be a positive number not greater than 65535. For a wildcard service, specify an asterisk (*) as the port number.

Note: For more information about the SSL and SSL_TCP service types, see Secure Sockets Layer (SSL) Acceleration.

To configure a wildcard service for each firewall by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a wildcard service for each firewall" as shown:
   - **Service Name**—name
   - **Server**—serverName
4. In **Protocol**, select **ANY**, and in **Port**, select ***.
5. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [**-destIP** <ip_addr|ipv6_addr|*>] [**-transparent** (**YES** | **NO**)]
- bind lb monitor <monitorName> <serviceName>

Example

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

Parameters for configuring a monitor

monitorName (Name)

The name of the monitor. This is a mandatory argument. Maximum Length: 31.

type (Type)

The type of monitor. This is a mandatory argument. Default: PING.

destIP (Destination IP)

The IP address to which the probe is sent. If the destination IP address is set to 0, the destination IP address is that of the server to which the monitor is bound. Default value: 0

transparent (Transparent)

The state of the monitor for transparent devices, such as firewalls, based on the responsiveness of the services behind them. If the monitoring of transparent devices is enabled, the destination IP address should be specified. The probe is sent to the specified destination IP address using the MAC address of the transparent device. Possible values: YES, NO. Default value: NO

To create and bind a transparent monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a monitor, as shown:
   - Name*
   - Type*—type
   - Destination IP
   - Transparent

   * A required parameter
4. Click **Create**, and then click **Close**. In the **Monitors** pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> ANY * *

Example

add lb vserver Vserver-LB-1 ANY * *

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a virtual server” as shown:
   - Name—name
4. In **Protocol**, select **ANY**, and in **IP Address** and **Port**, select ***.
5. Click **Create**, and then click **Close**. The virtual server you created appears in the **Load Balancing Virtual Servers** pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**m** <RedirectionMode>

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

Parameter for configuring the virtual server in MAC rewrite mode

m

The load balancing redirection mode. Possible Values: IP, MAC. Default: IP.

If set to IP, the destination IP address of the request is changed to the IP address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server.

If set to MAC, the destination MAC address is changed to the MAC address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server. With this setting, the destination IP address of the traffic is not changed.

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, under **Redirection Mode**, click **MAC-Based**.
4. Click **OK**.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <name> <serviceName>

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service (for example, **Vserver-LB-1**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Service-HTTP-1**).
5. Click **OK**.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

To configure a service for each virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

add service <name> <serverName> HTTP <port>

Example

add service Service-HTTP-1 10.102.29.5 HTTP 80

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously-created server object, or the IP address of the load-balanced server, that is associated with this service, in either IPv4 or IPv6 format.

When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

Note: For more information about the SSL and SSL_TCP service types, see Secure Sockets Layer (SSL) Acceleration.

To configure a service for each virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service” as shown:
   - **Service Name**—name
   - **Server**—serverName
   - **Port**—port
4. In **Protocol**, specify **HTTP**. Under **Available Monitors**, select **HTTP**.
5. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Configure a monitor for each service

To bind a monitor to a service by using the NetScaler command line

At the NetScaler command prompt, type:

bind mon <MonitorName> <ServiceName>

Example

bind mon monitor-HTTP-1 Service-HTTP-1

To bind a monitor to a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to bind the monitor (for example, **Service-HTTP-1**), and then click **Open**.
3. On the **Monitors** tab, in the **Available** list box, select the monitor you want to bind the service (for example, monitor-HTTP-1), and then click **Add**.
4. In the **Configured** box, click **OK**.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> HTTP <ip> <port>

Example

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a virtual server" as shown:
   - Name—name
   - IP Address—IPAddress

     Note: If the virtual server uses IPv6, select the **IPv6** check box and enter the address in IPv6 format (for example, **1000:0000:0000:0000:0005:0600:700a:888b**).
   - Port—port
4. Under **Protocol**, select **HTTP**.
5. Click **Create**, and then click **Close**. The virtual server you created appears in the **Load Balancing Virtual Servers** pane.

Bind HTTP services to the HTTP virtual server

To bind HTTP services to the wildcard virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <name> <serviceName>

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind HTTP services to the wildcard virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service (for example, **Vserver-LB-1**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Service-HTTP-1**).
5. Click **OK**.

Note: You can bind a service to multiple virtual servers.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save config
- show vserver <name>

Example

```
save config
show lb vserver FWLBVIP2
      FWLBVIP2 (*:*) - ANY    Type: ADDRESS
```

```
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click **Save**.
2. In the **Save Config** dialog box, click **Yes**.
3. In the navigation pane, click **Load Balancing**, and then click **Virtual Servers**.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the **Details** pane are correct.
5. In the navigation pane, click **Load Balancing**, and then click **Services**.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the **Details** pane are correct.

Monitoring a Firewall Load Balancing Setup in an Enterprise Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the NetScaler command line

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the NetScaler command prompt, type:

stat lb vserver [-**detail**] **[<name>]**

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
```

```
                  vsvrIP  port     Protocol       State    Req/s   Hits/s
One                  *    80         HTTP          UP      5/s      0/s
Two                  *     0          TCP        DOWN      0/s      0/s
Three                *  2598          TCP        DOWN      0/s      0/s
dnsVirtualNS    10.102.29.90    53          DNS        DOWN      0/s      0/s
BRVSERV          10.10.1.1    80         HTTP        DOWN      0/s      0/s
LBVIP          10.102.29.66    80         HTTP          UP      0/s      0/s
 Done
```

Parameters for displaying statistics

detail

Include the statistics for hits per second and the total number of hits.

name

Name of the virtual server whose statistics are displayed.

To display virtual server statistics by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.
3. In the details pane, click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the NetScaler command line

At the NetScaler command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service whose statistics you want to view (for example, **Service-HTTP-1**).
3. Click **Statistics**. The statistics appear in a new window.

Global Server Load Balancing

NetScaler appliances configured for global server load balancing (GSLB) provide for disaster recovery and ensure continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in case of an outage.

Following are some typical GSLB configurations:

- **Active-active data center setup**. Consists of multiple active data centers. Client requests are load balanced across active data centers.
- **Active-standby data center setup**. Consists of an active and a standby data center. When a failover occurs as a result of a disaster event, the standby data center becomes operational.
- **Proximity setup**. Directs client requests to the data center that is closest in geographical distance or network distance.

In a typical configuration, a local DNS server sends client requests to a GSLB virtual server, to which are bound GSLB services. A GSLB service identifies a load balancing or content switching virtual server, which can be at the local site or a remote site. If the GSLB virtual server selects a load balancing or content switching virtual server at a remote site, it sends the virtual server's IP address to the DNS server, which sends it to the client. The client then resends the request to the new virtual server at the new IP.

The GSLB entities that you must configure are the GSLB sites, the GSLB services, the GSLB virtual servers, load balancing or content switching virtual servers, and authoritative DNS (ADNS) services. You must also configure MEP. You can also configure DNS views to expose different parts of your network to clients accessing the network from different locations.

Note: To take full advantage of the NetScaler GSLB features, you should use NetScaler appliances for load balancing or content switching at each data center, so that your GSLB configuration can use the proprietary Metric Exchange Protocol (MEP) to exchange site metrics.

How GSLB Works

With ordinary DNS, when a client sends a domain name system (DNS) request, it receives a list of IP addresses of the domain or service. Generally, the client chooses the first IP address in the list and initiates a connection with that server. The DNS server uses a technique called DNS round robin to rotate through the IPs on the list, sending the first IP address to the end of the list and promoting the others after it responds to each DNS request. This technique ensures equal distribution of the load, but it does not support disaster recovery, load balancing based on load or proximity of servers, or persistence.

When you configure GSLB on NetScaler appliances and enable Metric Exchange Protocol (MEP), the appliances use the DNS infrastructure to connect the client to the data center that best meets the criteria that you set. The criteria can designate the least loaded data center, the closest data center, the data center that responds most quickly to requests from the client's location, a combination of those metrics, and SNMP metrics. An appliance keeps track of the location, performance, load, and availability of each data center and uses these factors to select the data center to which to send a client request.

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include GSLB sites, GSLB services, GSLB virtual servers, load balancing and/or content switching servers, and ADNS services.

How GSLB Works

With ordinary DNS, when a client sends a domain name system (DNS) request, it receives a list of IP addresses of the domain or service. Generally, the client chooses the first IP address in the list and initiates a connection with that server. The DNS server uses a technique called DNS round robin to rotate through the IPs on the list, sending the first IP address to the end of the list and promoting the others after it responds to each DNS request. This technique ensures equal distribution of the load, but it does not support disaster recovery, load balancing based on load or proximity of servers, or persistence.

When you configure GSLB on NetScaler appliances and enable Metric Exchange Protocol (MEP), the appliances use the DNS infrastructure to connect the client to the data center that best meets the criteria that you set. The criteria can designate the least loaded data center, the closest data center, the data center that responds most quickly to requests from the client's location, a combination of those metrics, and SNMP metrics. An appliance keeps track of the location, performance, load, and availability of each data center and uses these factors to select the data center to which to send a client request.

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include GSLB sites, GSLB services, GSLB virtual servers, load balancing and/or content switching servers, and ADNS services.

GSLB Sites

A typical GSLB setup consists of data centers, each of which has various network appliances that may or may not be NetScaler appliances. The data centers are called GSLB sites. Each GSLB site is managed by a NetScaler appliance that is local to that site. Each of these appliances treats its own site as the local site and all other sites, managed by other appliances, as remote sites.

If the appliance that manages a site is the only NetScaler appliance in that data center, the GSLB site hosted on that appliance acts as a bookkeeping placeholder for auditing purposes, because no metrics can be collected. Typically, this happens when the appliance is used only for GSLB, and other products in the data center are used for load balancing or content switching.

GSLB Services

A GSLB service is usually a representation of a load balancing or content switching virtual server, although it can represent any type of virtual server. The GSLB service identifies the virtual server's IP address, port number, and service type. GSLB services are bound to GSLB virtual servers on the NetScaler appliances managing the GSLB sites. A GSLB service bound to a GSLB virtual server in the same data center is local to the GSLB virtual server. A GSLB service bound to a GSLB virtual server in a different data center is remote from that GSLB virtual server.

GSLB Virtual Servers

A GSLB virtual server has one or more GSLB services bound to it, and load balances traffic among those services. It evaluates the configured GSLB methods (algorithms) to select the appropriate service to which to send a client request. Because the GSLB services can represent either local or remote servers, selecting the optimal GSLB service for a request has the effect of selecting the data center that should serve the client request.

The domain for which global server load balancing is configured must be bound to the GSLB virtual server, because one or more services bound to the virtual server will serve requests made for that domain.

Unlike other virtual servers configured on a NetScaler appliance, a GSLB virtual server does not have its own virtual IP address (VIP).

Load Balancing or Content Switching Virtual Servers

A load balancing or content switching virtual server represents one or many physical servers on the local network. Clients send their requests to the load balancing or content switching virtual server's virtual IP (VIP) address, and the virtual server balances the load across the physical servers. After a GSLB virtual server selects a GSLB service representing either a local or a remote load balancing or content switching virtual server, the client sends the request to that virtual server's VIP address.

For more information about load balancing or content switching virtual servers and services, see Load Balancing Load Balancing, or Content SwitchingContent Switching.

ADNS Services

An ADNS service is a special kind of service that responds only to DNS requests for domains for which the NetScaler appliance is authoritative. When an ADNS service is configured, the appliance owns that IP address and advertises it. Upon reception of a DNS request by an ADNS service, the appliance checks for a GSLB virtual server bound to that domain. If a GSLB virtual server is bound to the domain, it is queried for the best IP address to which to send the DNS response.

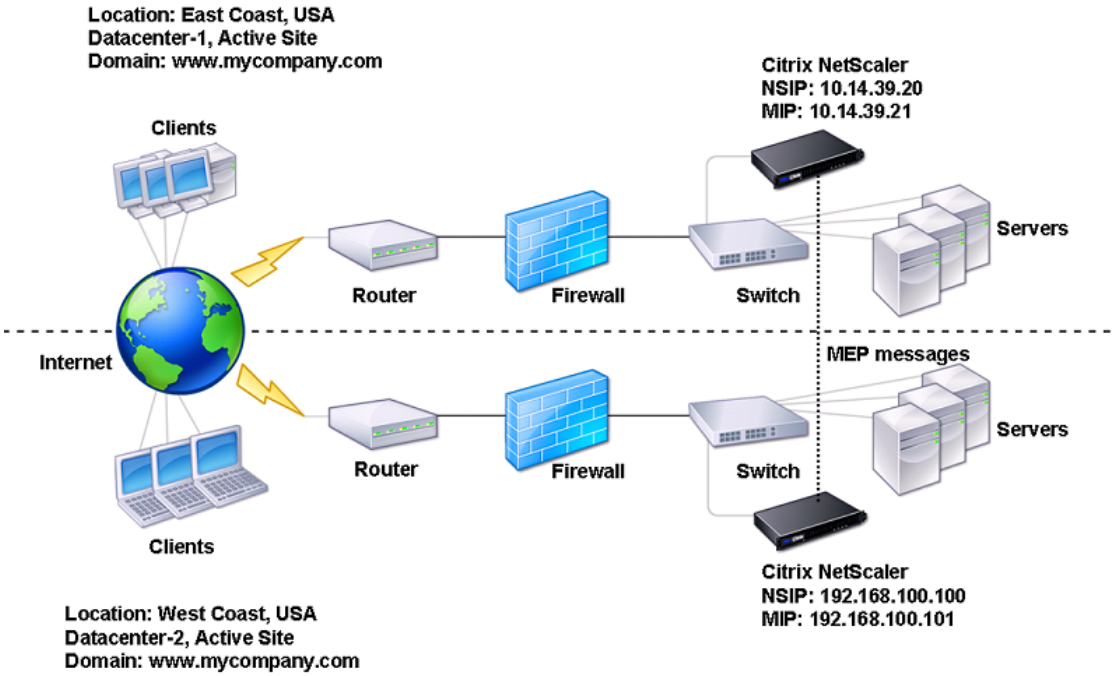
DNS VIPs

A DNS virtual IP is a virtual IP (VIP) address that represents a load balancing DNS virtual server on the NetScaler appliance. DNS requests for domains for which the NetScaler appliance is authoritative can be sent to a DNS VIP.

Configuring Global Server Load Balancing (GSLB)

Global server load balancing is used to manage traffic flow to a web site hosted on two separate server farms that ideally are in different geographic locations. For example, consider a Web site, www.mycompany.com, which is hosted on two geographically separated server farms or data centers. Both server farms use NetScaler appliances. The NetScaler appliances in these server farms are set up in one-arm mode and function as authoritative DNS servers for the www.mycompany.com domain. The following figure illustrates this configuration.

Figure 1. Basic GSLB Topology



To configure such a GSLB setup, you must first configure a standard load balancing setup for each server farm or data center. This enables you to balance load across the different servers in each server farm. Then, configure both NetScaler appliances as authoritative DNS (ADNS) servers. Next, create a GSLB site for each server farm, configure GSLB virtual servers for each site, create GLSB services, and bind the GSLB services to the GSLB virtual servers. Finally, bind the domain to the GSLB virtual servers. The GSLB configurations on the two appliances at the two different sites are identical, although each sites's load-balancing configuration is specific to that site.

Configuring a Standard Load Balancing Setup

A load balancing virtual server balances the load across different physical servers in the data center. These servers are represented as services on the NetScaler appliance, and the services are bound to the load balancing virtual server.

For details on configuring a basic load balancing setup, see Load Balancing.

Configuring an Authoritative DNS Service

When you configure the NetScaler appliance as an authoritative DNS server, it accepts DNS requests from the client and responds with the IP address of the data center to which the client should send requests.

Note: For the NetScaler to be authoritative, you must also create SOA and NS records. For more information about SOA and NS records, see Domain Name System.

To create an ADNS service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an ADNS service and verify the configuration:

- add service <name> <IPAddress> **ADNS** <port>
- show service <name>

Example

```
add service Service-ADNS-1 10.14.39.21 ADNS 53
show service Service-ADNS-1
```

To modify an ADNS service by using the NetScaler command line

At the NetScaler command prompt, type the following command:

set service <name> <IPAddress> **ADNS** <port>

Example

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
```

To remove an ADNS service by using the NetScaler command line

At the NetScaler command prompt, type the following command:

rm service <name>

Example

rm service Service-ADNS-1

Parameters for configuring an ADNS service

name

The name of the ADNS service you are creating. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

IPAddress

The IP address of the server that the ADNS service represents. You can configure the ADNS service to use a mapped IP address (MIP), subnet IP address (SNIP), or any new NetScaler-owned IP address.

port

The port on which the service communicates with the application on the server. This number must correspond to the protocol that the application supports. The port number must always be a positive number not exceeding 65535.

To configure an ADNS service by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Services.**
2. In the details pane, do one of the following:
   - To create a new service, click **Add.**
   - To modify an existing service, select the service, and then click **Open.**
3. In the **Create Service** or **Configure Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring an ADNS service” as shown:
   - **Service Name*****—name
   - **Service Name*****—name
   - **Protocol*****—(Select ADNS as the protocol.)
   - **Port*****—port
4. Click **Create** or **OK**, and then click **Close**. The server that you created appears in the GSLB Services pane.

Configuring a Basic GSLB Site

A GSLB site is a representation of a data center in your network and is a logical grouping of GSLB virtual servers, services, and other network entities. Typically, in a GSLB set up, there are many GSLB sites that are equipped to serve the same content to a client. These are usually geographically separated to ensure that the domain is active even if one site goes down completely. All of the sites in the GSLB configuration must be configured on every NetScaler appliance hosting a GSLB site. In other words, at each site, you configure the local GSLB site and each remote GSLB site.

Once GSLB sites are created for a domain, the NetScaler appliance sends client requests to the appropriate GSLB site as determined by the GSLB algorithms configured.

To create a GSLB site by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a GSLB site and verify the configuration:

- add gslb site <siteName> <siteIPAddress>
- show glsb site <siteName>

Example

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
```

To modify or remove a GSLB Site by using the NetScaler command line

- To modify a GSLB site, use the set gslb site command, which is just like using the add gslb site command, except that you enter the name of an existing GSLB Site.
- To unset a site parameter, use the unset gslb site command, followed by the siteName value and the name of the parameter to be reset to its default value.
- To remove a GSLB site, use the rm gslb site command, which accepts only the <name> argument.

Parameters for configuring a GSLB site

siteName

A name for the data center you are adding as a GSLB site. This alphanumeric string is required and cannot be changed after the site is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

siteIPAddress

The IP address of the GSLB site. This IP address is a system-owned IP address. You can use any IP address configured as a SNIP, MIP, or GSLB site IP address. This is a mandatory parameter.

Note: To avoid a site going down during an HA failover event in a GSLB setup with an independent network configuration high availability deployment, the GSLB site IP address must be on the same subnet as the virtual IP (VIP) address of the load balancing or content switching virtual server that is bound to the service(s) provided by that GSLB site. In an independent network configuration high availability deployment, two nodes do not share the same subnet IPs (SNIPs) or mapped IPs (MIPs), but they have common VIPs.

To configure a basic GSLB site by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Sites**.
2. In the details pane, do one of the following:
   - To create a new site, click **Add**.
   - To modify an existing site, select the site, and then click **Open**.
3. In the **Create GSLB Site** or **Configure GSLB Site** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a GSLB site” as shown:
   - **Name*** —siteName
   - **Site IP Address*** —siteIPAddress

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. The GSLB site you created appears in the GSLB Sites pane.

To view the statistics of a GSLB site by using the NetScaler command line

At the NetScaler command prompt, type:

stat gslb site <siteName>

Example

stat gslb site Site-GSLB-East-Coast

To view the statistics of a GSLB site by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Sites.**
2. In the **GSLB Sites** pane, select the GSLB site whose statistics you want to view.
3. Click **Statistics**.

Configuring a GSLB Service

A GSLB service is a representation of a load balancing or content switching virtual server. A local GSLB service represents a local load balancing or content switching virtual server. A remote GSLB service represents a load balancing or content switching virtual server configured at one of the other sites in the GSLB setup. At each site in the GSLB setup, you can create one local GSLB service and any number of remote GSLB services.

Creating GSLB Services

To create a GSLB service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a GSLB service and verify the configuration:

- add gslb service <serviceName> <serverName | serverIP> <serviceType> <port>**-siteName** <string>
- show gslb service <serviceName>

Example

add gslb service Service-GSLB-1 10.14.39.14 HTTP 80 –siteName Site-GSLB-East-Coast
show gslb service Service-GSLB-1

To modify or remove a GSLB service by using the NetScaler command line

- To modify a GSLB service, use the set gslb service <serviceName> command. For this command, specify the name of the GSLB service whose configuration you want to modify. You can change the existing values of the parameters either specified by you or set by default. You can change the value of more than one parameter in the same command. Refer to the add gslb service command for details about the parameters. Example

  ```
  > set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -maxClient 8
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 25 kbits
  ```

- To reset a parameter to its default value, you can use the unset gslb service <serviceName> command and the parameters to be unset. Example

  ```
  > unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 0 kbits
  ```

- To remove a GSLB service, use the rm gslb service <serviceName> command.

Parameters for configuring a GSLB service

serviceName (Service Name)

The name of the service being configured. This alphanumeric string is required. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

siteName (Site Name)

The name of the GSLB site that this service represents.

serviceType (Service Type)

The type of service or protocol used in client requests. Possible values: HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, MSSQL, MYSQL, RADIUS, RDP, SIP_UDP, RTSP.

port (Port)

Port number on which the service runs.

serverName or ipAddress (Server IP)

The server name or IP address of the GSLB service being configured. Must be the same as the virtual IP (VIP) address of a local or remote load balancing or content switching virtual server.

publicIP (Public IP)

The public IP address of the NAT translator for a GSLB service that is on a private network.

To create a GSLB service by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Services**.
2. In the details pane, do one of the following:
   - To create a new service, click **Add**.
   - To modify an existing service, select the service, and then click **Open**.
3. In the **Create GSLB Service** or **Configure GSLB Service** dialog box, set the following parameters:
   - **Service Name***
   - **Site Name***
   - **Server Name** - The servers added to the NetScaler configuration are displayed in a dropdown list. If you want to add a new server, click **New...**, and then in the **Create Server** dialog box, type the necessary details. For more information about creating servers, see Adding a Server.
   - **Service Type**
   - **Port**

   Note: In the Site Name and Server Name lists, the most recently used value is displayed as selected. Make sure that you select the site and server you want to specify.
4. Click **Create**, and then click **Close**. The GSLB service you created appears in the **GSLB Services** pane.

To view the statistics of a GSLB service by using the NetScaler command line

At the NetScaler command prompt, type:

stat gslb service <serviceName>

Example

stat gslb service Service-GSLB-1

To view the statistics of a GSLB service by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Services** pane, select the GSLB Service whose statistics you want to view.
3. Click **Statistics**.

Enabling and Disabling GSLB Services

Before you use a GSLB service for load balancing, it must be enabled. If the service is disabled, it is not included in load balancing even though it exists on the NetScaler appliance.

To enable or disable a GSLB service by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable service <name>
- disable service <name>

Example

```
> enable service Service-GSLB-1
 Done
> disable service Service-GSLB-1
 Done
```

To enable or disable a GSLB service by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Services**.
2. In the **GSLB Services** pane, select the GSLB service which you want to enable or disable.
3. Click **enable** or **disable**.

Configuring a GSLB Virtual Server

A GSLB virtual server is an entity that represents one or more GSLB services and balances traffic between them. It evaluates the configured GSLB methods or algorithms to select a GSLB service to which to send the client request.

Creating GSLB Virtual Servers

To create a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a GSLB virtual server and verify the configuration:

- add gslb vserver <name> <serviceType> **-ipType** (IPv4 | IPv6)
- show gslb vserver <name>

Example

```
add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
show gslb vserver Vserver-GSLB-1
show gslb vserver Vserver-GSLB-2
```

To modify or remove a GSLB virtual server by using the NetScaler command line

- To modify a GSLB virtual server, use the set gslb vserver command, which is just like using the add gslb vserver command, except that you enter the name of an existing GSLB virtual server.
- To reset a parameter to is default value, you can use the unset gslb vserver command followed by the vserverName value and the name of the parameter to be unset.
- To remove a GSLB virtual server, use the rm gslb vserver command, which accepts only the <name> argument.

Parameters for configuring a GSLB virtual server

name

The name of the GSLB virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serviceType

The service type of the virtual server, that is, the type of content in the processed requests. Possible values: HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RDP, ANY.

ipType

Specifies whether this virtual server supports services that use the IPv4 or IPv6 protocol for IP addresses. Possible values: IPv4, IPv6. Default: IPv4.

To create a GSLB virtual server by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the details pane, do one of the following:
   - To create a new GSLB virtual server, click **Add.**
   - To modify an existing GSLB virtual server, select the service, and then click **Open**.
3. In the **Create GSLB Virtual Server** or **Configure GSLB Virtual Server** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a GSLB virtual server" as shown:
   - **Name***—name
   - **Service Type***—serviceType
   - **IPv6**—ipType (To specify IPv6, select the check box. For IPv4, clear the check box.)

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. The GSLB virtual server that you created appears in the GSLB Virtual Servers pane.

To view the statistics of a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

stat gslb vserver <name>

Example

stat gslb vserver Vserver-GSLB-1

To view the statistics of a GSLB virtual server by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the details pane, select the GSLB virtual server whose statistics you want to view.
3. Click **Statistics**.

Statistics of a GSLB service

When you run the stat gslb service command from the command line or click on the **Statistics** link from the configuration utility, the following details of the service will be displayed:

- **Request bytes.** Total number of request bytes received on this service or virtual server.
- **Response bytes.** Number of response bytes received by this service or virtual server.
- **Current client established connections.** Number of client connections in ESTABLISHED state.
- **Current load on the service.** Load on the service (Calculated from the load monitor bound to the service).

The data of number of requests and responses, and the number of current client and server connections may not be displayed or may not be synchronized with the data of the corresponding load balancing virtual server.

Enabling and Disabling GSLB Virtual Servers

When you create a GSLB virtual server, it is enabled by default. If you disable it, it cannot process traffic. A disabled GSLB virtual server is not included in GSLB configuration but is not removed from the NetScaler appliance.

To enable or disable a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable gslb vserver <GSLBVserverName>
- disable gslb vserver <GSLBVserverName>

Example

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
```

To enable or disable a GSLB virtual server by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the details pane, select the GSLB virtual server that you want to enable or disable.
3. Click **enable** or **disable**.

Binding GSLB Services to a GSLB Virtual Server

Once the GSLB services and virtual server are configured, relevant GSLB services must be bound to the GSLB virtual server to activate the configuration.

To bind a GSLB service to a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a GSLB service to a GSLB virtual server and verify the configuration:

- bind gslb vserver <name> **-serviceName** <string>
- show gslb vserver <name>

Example

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
```

To unbind a GSLB service from a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

unbind gslb vserver <name> **-serviceName** <string>

To bind GSLB services by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the details pane, select the GSLB Virtual Server to which you want to bind the services (for example, Vserver-GSLB-1).
3. Click **Open.**
4. On the **Services** tab, in the **Active** column, select the check boxes next to the GSLB services that you want to bind to the GSLB virtual server.
5. Click **OK.**

Binding a Domain to a GSLB Virtual Server

To make a NetScaler appliance the authoritative DNS server for a domain, you must bind the domain to the GSLB virtual server. When you bind a domain to a GSLB virtual server, the NetScaler adds an address record for the domain, containing the name of the GSLB virtual server. The start of authority (SOA) and name server (NS) records for the GSLB domain must be added manually.

For details on configuring SOA and NS records, see Domain Name System.

To bind a domain to a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a domain to a GSLB virtual server and verify the configuration:

- bind gslb vserver <name> **-domainName** <string>
- show gslb vserver <name>

Example

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
```

To unbind a GSLB domain from a GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

unbind gslb vserver <name> **-domainName** <string>

To bind a domain to a GSLB virtual server by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In **GSLB Virtual Servers** pane, select the GSLB Virtual Server to which you want to bind the domain (for example, Vserver-GSLB-1) and click **Open**.
3. In the **Configure GSLB Virtual Server** dialog box, on the **Domains** tab, do one of the following:
   - To create a new Domain, click **Add.**
   - To modify an existing Domain, select the domain, and then click **Open**.
4. In the **Create GSLB Domain** or **Configure GSLB Domain** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for Binding or Unbinding a Domain to a GSLB Virtual Server" as shown:
   - **Domain Name***—domainName (for example, **www.mycompany.com**)

   * A required parameter
5. Click **Create**.
6. Click **OK**.

To view the statistics of a domain by using the NetScaler command line

At the NetScaler command prompt, type:

stat gslb domain <name>

Example

stat gslb domain www.mycompany.com

Note: To view statistics for a particular GSLB domain, enter the name of the domain exactly as it was added to the NetScaler appliance. If you do not specify the domain name, or if you specify an incorrect domain name, statistics for all configured GSLB domains are displayed.

To view the statistics of a domain by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In **GSLB Virtual Servers** pane, select the GSLB Virtual Server (for example, Vserver-GSLB-1) and click **Open**.
3. In the **Configure GSLB Virtual Server** dialog box, on the **Domains** tab, select the doamin, and then click **Statistics**.

Synchronizing a Configuration in a GSLB Setup

Typically, a GSLB setup has a few data centers with a GSLB site configured for each data center. In each NetScaler, participating in GSLB, configure one GSLB site as a local site and the others as remote sites.

When you add another GSLB site at a later point of time, ensure that all the GSLB sites have the same configuration. To have the same configuration on all the GSLB sites, you can use the NetScaler appliance's GSLB configuration synchronization option.

The NetScaler appliance from which you use the synchronization option is referred to as the 'master node' and the GSLB sites on which the configuration is copied as 'slave nodes'. When you synchronize a GSLB configuration, the configurations on all the GSLB sites participating in the GSLB setup are made similar to that on the master node.

Synchronization is carried out in the following manner:

- When you synchronize a GSLB configuration (may also be referred to as 'auto sync'), the NetScaler finds the differences between the configuration of the master node and slave node, and changes the configuration of the slave node to make it similar to the master node.

  If you force a synchronization (use the 'force sync' option), the NetScaler deletes the GSLB configuration from the slave node and then configures the slave to make it similar to the master node.

- During synchronization, if a command fails, synchronization is not aborted.

- Synchronization is done only on the parent sites. If a GSLB site is configured as a child site, its configuration is not affected by synchronization.

Limitations of synchronization:

- On the master node, the names of the remote GSLB sites must be identical to the names of sites configured on the NetScaler appliances hosting those sites.

- During the synchronization, traffic disruptions may occur.

- NetScaler can synchronize only up to 80000 lines of the configuration.

- Synchronization may fail if the spill over method is changed from **CONNECTION** to **DYNAMIC CONNECTION**.

- If you interchange the site prefix of the GSLB services bound to a GSLB virtual server on the master node and then try to synchronize, synchronization may fail.

Note: To overcome the limitations due to some settings in the GSLB configuration, you can use the **force sync** option.

Before you start the synchronization of a GSLB setup, make sure that:

- On all the GSLB sites including the master node, management access should be enabled for the IP address of the corresponding GSLB site. The IP address of a GSLB site must be an IP address owned by the NetScaler.

  For more information about adding the GSLB site IP addresses and enabling Management Access, see Configuring a Basic GSLB Site and Configuring NetScaler-Owned IP Addresses.

- The GSLB configuration on the NetScaler appliance that is considered as the master node is complete and appropriate to be copied on all the sites.

Important: After a GSLB configuration is synchronized, the configuration cannot be rolled back on any of the GSLB sites. If some commands fail and some commands succeed, the successful commands cannot be rolled back.

To synchronize a GSLB configuration by using the NetScaler command line

At the NetScaler command prompt, type the following commands to synchronize and view the result of synchronization:

- sync gslb config [**-preview**] [**-forceSync** <string>] [**-debug**]
- show gslb syncStatus

Parameters for synchronizing a GSLB configuration

<no option>

If no option is passed to this command, synchronization happens on all the GSLB sites in the auto sync method.

preview (Preview)

Displays the commands that are executed on the slave nodes when the sync command is executed from the master node or when you click Run in the Synchronize GSLB Configuration dialog box. This command does not initiate the synchronization.

debug (Debug)

Displays a more verbose output of synchronization activity. This command initiates the synchronization.

forceSync <string> (Force Sync)

Synchronizes the configuration of the master node to a slave node. <string> indicates the slave node on which synchronization is to be done. If you specify forceSync, the string is mandatory.

In the configuration utility, when you select the **Synchronization Option** as **Force Sync**, the GSLB Site Name dropdown list is enabled so that you can select a GSLB site.

To synchronize the master node's configuration on all the slave nodes, set the forceSync string as ' all-sites ' or select **All Sites** in the **GSLB Site Name** list.

Note: If you select the force sync option, the synchronization starts without displaying the commands that are going to be executed.

To synchronize a GSLB configuration by using the configuration utility

1. In the navigation pane, click **GSLB**.
2. In the **GSLB** pane, under **GSLB Configuration**, click **Synchronize configuration on remote sites**.
3. In the **Synchronize GSLB Configuration** dialog box, select a synchronization option.
4. In the **Output** pane, you can see all the commands that are executed and the result of execution. If you want to save the output into a file on the local system or on a system on the network, click the **Save output text to a file** link.
5. Click **Run**.
6. Click **Close**.

Viewing and Configuring a GSLB Setup by Using the GSLB Visualizer

The configuration utility includes a GSLB Visualizer tool, which provides an alternative way to view and configure entities in a GSLB configuration. The visualizer displays all configured GSLB domains, GSLB services, GSLB sites, ADNS services, and any monitors that are bound to the services. It also displays all the load balancing, content switching, cache redirection, and Access Gateway virtual servers that the GSLB services represent.

If you want to view the configurations of remote GSLB sites, you must configure the sites with public IP addresses and enable management access for each of them.

You can use the GSLB Visualizer to perform the following GSLB configuration tasks:

- Add, view, and configure GSLB domains and GSLB services.
- View and configure GSLB sites and ADNS services for each site.
- View and configure any monitors that are bound to the services.
- View and configure the content switching, load balancing, cache redirection, or Access Gateway virtual server that each GSLB service represents.
- View statistics for GSLB domains, sites, ADNS services, and virtual servers.
- View configuration details of any displayed entity.
- View load balancing and content switching virtual servers.
- View bindings for GSLB services, ADNS services, monitors, and virtual servers.
- Enable and disable GSLB services, ADNS services, monitors, and virtual servers.
- Copy the properties of any displayed entity to a document or spreadsheet.
- Remove a domain from the GSLB setup.
- Save the visual representation of the GSLB setup as an image.

To open the Visualizer and locate an entity

1. In the navigation pane, click **GSLB**.
2. In the details pane, under **Getting Started**, click **GSLB Visualizer**, and then do the following.
   - To pan the view of the displayed image, click as blank area of the image, hold down the mouse button, and drag the image.
   - To adjust the viewable area click **Zoom In** to increase or **Zoom Out** to decrease the size of the objects. You can readjust the viewable area by clicking **Best Fit**.
   - To locate a specific item, begin typing the item's name in the **Search** field. Entities whose names match the typed characters are highlighted. Continue typing until the item is uniquely identified. To clear the **Search** field, click the x adjacent to the field.

To add a GSLB domain and/or configure GSLB services and sites for the domain

1. Open the GSLB Visualizer and click **Domain**. Alternatively, if domains already exist in the GSLB setup, click the name of an existing domain.
2. Under **Related Tasks**, click **Add**.
3. Follow the instructions in the GSLB Wizard to add a GSLB domain and configure GSLB services and sites for the domain.

To view the configuration details of an entity

Open the GSLB Visualizer and do one of the following:

- To view a brief summary of an entity, place the pointer on the entity. A brief summary of the entity appears at the bottom of the viewable area.
- To view the detailed configuration information of the entity, click the entity. The configuration details for that entity appear in the **Details** area.

To modify a GSLB domain, site, service, monitor, or ADNS service

Open the GSLB Visualizer and do one of the following:

- Click the entity that you want to modify. Then, under **Related Tasks**, click **Open**.
- Double-click the entity that you want to modify.

- Right-click the entity that you want to modify, and then click **Open**. (This option is not available for GSLB sites.)

To view the entities to which a GSLB service, ADNS service, monitor, or virtual server is bound

Open the GSLB Visualizer and do one of the following:

- Click the entity whose binding information you want to view, and then, under **Related Tasks**, click **Show Bindings**.
- Right-click the entity, and then click **Show Bindings**.

To view the Visualizer for load balancing and content switching virtual servers from the GSLB Visualizer

Open the GSLB Visualizer and do one of the following:

- Click the load balancing or content switching virtual server whose Visualizer you want to view, and then, under **Related Tasks**, click **Visualizer**.
- Right-click the virtual server, and then click **Visualizer**.

To view statistics for a GSLB service, site, ADNS service, or virtual server

Open the GSLB Visualizer and do one of the following:

- Click the entity whose statistics you want to view, and then, under **Related Tasks**, click **Statistics**.
- Right-click the entity whose statistics you want to view, and then click **Statistics**. (This option is not available for GSLB sites.)

To enable or disable a GSLB service, ADNS service, monitor, or virtual server

Open the GSLB Visualizer and do one of the following to enable or disable the entity:

- To enable the entity, click the entity and, under **Related Tasks**, click **Enable**. Alternatively, right-click the entity that you want to enable, and then click **Enable**.
- To disable the entity, click the entity and, under **Related Tasks**, click **Disable**. Alternatively, right-click the entity that you want to disable, and then click **Disable**.

To copy the properties of an entity to a document or spreadsheet

Open the GSLB Visualizer and do one of the following:

- Click the entity whose properties you want to copy, and then, under **Related Tasks,** click **Copy Properties**.
- Right-click the entity, and then click **Copy**. (This option is not available for GSLB sites.)

To save the visual representation of the GSLB setup as an image

1. Open the GSLB Visualizer.
2. If necessary, adjust the viewable area by using the **Best Fit**, **Zoom In**, and **Zoom Out** buttons.
3. Click **Save Image**.
4. In the **Save Graph Image** dialog box, browse to the folder in which you want to save the image.
5. In **File Name** text box, type the name, and then click **Save**.

To remove a domain from the GSLB setup

1. Open the GSLB Visualizer and do one of the following:
   - Click the domain that you want to remove, and then, under **Related Tasks**, click **Remove**.
   - Right-click the domain, and then click **Remove**.
2. Under **Remove?**, click **Yes**.

Configuring the Metrics Exchange Protocol (MEP)

The data centers in a GSLB setup exchange metrics with each other through the metrics exchange protocol (MEP), which is a proprietary protocol for the Citrix NetScaler. The exchange of the metric information begins when you create a GSLB site. These metrics comprise load, network, and persistence information.

MEP is required for health checking of data centers to ensure their availability. A connection for exchanging network metrics can be initiated by either of the data centers involved in the exchange, but a connection for exchanging site metrics is always initiated by the data center with the lower IP address. By default, the data center uses a subnet IP address (SNIP) or a mapped IP address (MIP) to establish a connection to the IP address of a different data center. However, you can configure a specific SNIP, MIP, the NetScaler IP address (NSIP), or a virtual IP address (VIP) as the source IP address for metrics exchange. The communication process between GSLB sites uses TCP port 3011, so this port must be open on firewalls that are between the NetScaler appliances.

Note: You cannot configure a GSLB site IP address as the source IP address for site metrics exchange.

If the source and target sites for a MEP connection (the site that initiates a MEP connection and the the site that receives the connection request, respectively) have both private and public IP addresses configured, the sites exchange MEP information by using the public IP addresses.

You can also bind monitors to check the health of remote services. When monitors are bound, metric exchange does not control the state of the remote service. If a monitor is bound to a remote service and metrics exchange is enabled, the monitor controls the health status. Binding the monitors to the remote service allows the NetScaler to interact with a non-NetScaler load balancing device. The NetScaler can monitor non-NetScaler devices but cannot perform load balancing on them. The NetScaler can monitor non-NetScaler devices, and can perform load balancing on them if monitors are bound to all GSLB services and only static load balancing methods (such as the round robin, static proximity, or hash-based methods) are used.

Configuring Site Metric Exchange

Site metrics exchanged between the GSLB sites include the status of each load balancing and content switching virtual server, the current number of connections, the current packet rate, and current bandwidth usage information.

The NetScaler appliance needs this information to perform load balancing between the sites. The site metric exchange interval is 1 second. A remote GSLB service must be bound to a local GSLB virtual server to enable the exchange of site metrics with the remote service.

To enable or disable site metric exchange by using the NetScaler command line

At a NetScaler command prompt, type the following commands to enable or disable site metric exchange and verify the configuration:

- set gslb site <GSLBSiteName> **-metricExchange**(ENABLED|DISABLED)
- show gslb site <GSLBSiteName>

Example

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
```

To enable or disable site metric exchange by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Sites.**
2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, select or clear the check box next to the **Metric Exchange** and click **OK**.

Configuring Network Metric Information Exchange

You can enable or disable the exchange of round trip time (RTT) information about the client's local DNS when the GSLB dynamic method (RTT) is enabled. This information is exchanged every 5 seconds.

For details about changing the GSLB method to a method based on RTT, see Changing the GSLB Method.

To enable or disable network metric information exchange by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable or disable network metric information exchange and verify the configuration:

- set gslb site <GSLBSiteName> **-nwmetricExchange** (ENABLED|DISABLED)
- show gslb site <GSLBSiteName>

Example

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
```

To enable or disable network metric information exchange by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Sites.**
2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, select or clear the check box next to the **Network Metric Exchange** and click **OK**.

Configuring Persistence Information Exchange

You can enable or disable the exchange of persistence information at each site. This information is exchanged every 5 seconds between NetScaler appliances participating in GSLB.

For details about configuring persistence, see Configuring Persistent Connections.

To enable/disable persistence information exchange by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable or disable persistence information exchange and verify the configuration:

- set gslb site <GSLBSiteName> **-sessionExchange** (ENABLED|DISABLED)
- show gslb site <GSLBSiteName>

Example

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
```

To enable/disable persistence information exchange by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Sites.**
2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, select or clear the check box next to the **Persistence Session Entry Exchange** and click **OK**.

Configuring Site-to-Site Communication

GSLB site-to-site communication is between the remote procedure call (RPC) nodes that are associated with the communicating sites. A master GSLB site establishes connections with slave sites to synchronize GSLB configuration information and to exchange site metrics.

An RPC node is created automatically when a GSLB site is created, and is assigned an internally generated user name and password. The NetScaler appliance uses this user name and password to authenticate itself to remote GSLB sites during connection establishment. No configuration steps are necessary for an RPC node, but you can specify a password of your choice, enhance security by encrypting the information that GSLB sites exchange, and specify a source IP address for the RPC node.

The appliance needs a NetScaler-owned IP address to use as the source IP address when communicating with other GSLB sites. By default, the RPC nodes use either a subnet IP (SNIP) address or a mapped IP (MIP) address, but you might want to specify an IP address of your choice.

Changing the Password of an RPC Node

You can secure the communication between sites in your GSLB setup by changing the password of each RPC node. After you change the password for the RPC node of the local site, you must manually propagate the change to the RPC node at each of the remote sites.

The password is stored in encrypted form. You can verify that the password has changed by using the `show rpcNode` command to compare the encrypted form of the password before and after the change.

To change the password of an RPC node by using the NetScaler command line

At the NetScaler command line, type the following commands to change the password of an RPC node:

- set ns rpcNode <IPAddress> {-password}
- show rpcNode

Example

```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
   SrcIP:  *             Secure:  OFF
Done
>
```

Parameters for changing the password of an RPC node

IPAddress

The IP address of the GSLB site to which the RPC node belongs. This is the value of the `IPAddress` field in the output of the `show rpcNode` CLI command.

password

The password that the RPC node must use to authenticate itself to other nodes in the GSLB configuration. By default, a password is configured for all RPC nodes. If you change the password for an RPC node, make sure you propagate that change to the RPC node at each of the other sites. Maximum length: 31 characters.

To change the password of an RPC node by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **RPC**.
2. In the details pane, click the RPC node for which you want to change the password, and then click **Open**.
3. In the **Configure RPC Node** dialog box, in **Password** and **Confirm Password**, specify the password that you want the RPC node to use.

Encrypting the Exchange of Site Metrics

You can secure the information that is exchanged between GSLB sites by setting the secure option for the RPC nodes in the GSLB setup. With the secure option set, the NetScaler appliance encrypts all communication sent from the node to other RPC nodes.

To encrypt the exchange of site metrics by using the NetScaler command line

At the NetScaler command prompt, type the following commands to encrypt the exchange of site metrics and verify the configuration:

- set ns rpcNode <IPAddress> [**-secure** (YES | NO)]
- show rpcNode

Example

```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3) IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3 Secure: ON
 Done
>
```

Parameters for encrypting the exchange of site metrics

IPAddress

The IP address of the GSLB site to which the RPC node belongs. This is the value of the `IPAddress` field in the output of the `show rpcNode` CLI command.

Secure (Secure)

Encrypt all communication sent from the RPC node. Possible values: YES, NO. Default: NO.

To encrypt the exchange of site metrics by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **RPC**.
2. In the details pane, click the RPC node whose communication you want to encrypt, and then click **Open**.
3. In the **Configure RPC Node** dialog box, click **Secure**.
4. Click **OK**.

Configuring the Source IP Address for an RPC Node

By default, the NetScaler appliance uses a NetScaler-owned subnet IP (SNIP) address or mapped IP (MIP) address as the source IP address for an RPC node, but you can configure the appliance to use a specific SNIP address or MIP address. If neither a SNIP address nor a MIP address is available, the GSLB site cannot communicate with other sites. In such a scenario, you must configure either the NetScaler IP (NSIP) address or a virtual IP (VIP) address as the source IP address for an RPC node. A VIP address can be used as the source IP address of an RPC node only if the RPC node is a remote node. If you configure a VIP address as the source IP address and remove the VIP address, the appliance uses a SNIP address or a MIP address.

To specify a source IP address for an RPC node by using the NetScaler command line

At the NetScaler command prompt, type the following commands to change the source IP address for an RPC node and verify the configuration:

- set ns rpcNode <IPAddress> [**-srcIP** <ip_addr|ipv6_addr|*>]
- show ns rpcNode

Example

```
> set rpcNode 192.0.2.4 -srcIP 192.0.2.3
 Done
> show rpcNode
.
.
.
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:  192.0.2.3       Secure:  OFF
 Done
>
```

Parameters for specifying a source IP address of an RPC node

IPAddress

The IP address of the GSLB site to which the RPC node belongs. This is the value of the `IPAddress` field in the output of the `show rpcNode` CLI command.

srcIP

The subnet IP (SNIP) address, mapped IP (MIP) address, NetScaler IP (NSIP) address, or virtual IP (VIP) address that you want the appliance to use as the source IP address for exchanging site metrics. By default, the appliance uses a SNIP address or a MIP address, but you can configure the node to use a SNIP address or MIP address of your choice, or the NSIP address. For a remote node, you also have the option of configuring a VIP address as the source IP address. If neither a SNIP address nor a MIP address is available, and you have not configured a source IP address, a GSLB site cannot exchange site metrics with other sites. The default setting is an asterisk (*), which indicates that the default setting (SNIP address or MIP address) is being used.

To specify a source IP address for an RPC node by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **RPC**.
2. In the details pane, click the RPC node for which you want to assign a specific source IP address for site metrics exchange, and then click **Open**.
3. In the **Configure RPC Node** dialog box, in **Source IP Address**, enter the IP address that you want the RPC node to use as the source IP address.

Customizing Your GSLB Configuration

Once your basic GSLB configuration is operational, you can customize it by modifying the bandwidth of a GSLB service, configuring CNAME based GSLB services, static proximity, dynamic RTT, persistent connections, or dynamic weights for services, or changing the GSLB Method.

You can also configure monitoring for GSLB services to determine their states.

These settings depend on your network deployment and the types of clients you expect to connect to your servers.

Modifying Maximum Connections or Maximum Bandwidth for a GSLB Service

You can restrict the number of new clients that can simultaneously connect to a load balancing or content switching virtual server by configuring the maximum number of clients and/or the maximum bandwidth for the GSLB service that represents the virtual server.

To modify the maximum clients or bandwidth of a GSLB service by using the NetScaler command line

At the NetScaler command prompt, type the following command to modify the maximum number of client connections or the maximum bandwidth of a GSLB service and verify the configuration:

- set gslb service <serviceName> [**-maxClients** <positive_integer>] [**-maxBandwidth** <positive_integer>]
- show gslb service <serviceName>

Example

```
set glsb service Service-GSLB-1 –maxBandwidth 100 –maxClients 100
show gslb service Service-GSLB-1
```

Parameters for modifying the maximum clients or bandwidth of a GSLB service

maxClients

The maximum number of simultaneous client connections that the GSLB service can handle.

maxBandwidth

The maximum bandwidth, in kbps, that a GSLB service can handle.

To modify the maximum clients or bandwidth of a GSLB service by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Services**.
2. In the details pane, select the service to be modified and click **Open.**
3. In the **Configure GSLB Service** dialog box specify values for one or both of the following parameters, which correspond to parameters described in “Parameters for modifying the maximum clients or bandwidth for a service” as shown:
   - **Max Clients**—maxClients
   - **Max Bandwidth**—maxBandwidth
4. Click **OK**.
5. Verify that the **Details** area displays the values that you entered.

Creating CNAME-Based GSLB Services

To configure a GSLB service, you can use the IP address of the server or a canonical name of the server. If you want to run multiple services (like an FTP and a Web server, each running on different ports) from a single IP address or run multiple HTTP services on the same port, with different names, on the same physical host, you can use canonical names (CNAMES) for the services.

For example, you can have two entries in DNS as ftp.example.com and www.example.com for FTP services and HTTP services on the same domain, example.com. CNAME-based GSLB services are useful in a multilevel domain resolver configuration or in multilevel domain load balancing. Configuring a CNAME-based GSLB service can also help if the IP address of the physical server is likely to change.

If you configure CNAME-based GSLB services for a GSLB domain, when a query is sent for the GSLB domain, the NetScaler appliance provides a CNAME instead of an IP address. If the A record for this CNAME record is not configured, the client must query the CNAME domain for the IP address. If the A record for this CNAME record is configured, the NetScaler provides the CNAME with the corresponding A record (IP address). The NetScaler appliance handles the final resolution of the DNS query, as determined by the GSLB method. The CNAME records can be maintained on a different NetScaler appliance or on a third-party system.

In an IP-address-based GSLB service, the state of a service is determined by the state of the server that it represents. However, a CNAME-based GSLB service has its state set to UP by default; the virtual server IP (VIP) address or metric exchange protocol (MEP) are not used for determining its state. If a desktop-based monitor is bound to a CNAME-based GSLB service, the state of the service is determined according to the result of the monitor probes.

You can bind a CNAME-based GSLB service only to a GSLB virtual server that has the **DNS Record Type** as CNAME.

The following are some of the features supported for a CNAME-based GSLB service:

- GSLB-policy based site affinity is supported, with the CNAME as the preferred location.
- Source IP persistence is supported. The persistency entry contains the CNAME information instead of the IP address and port of the selected service.

The following are the limitations of CNAME-based GSLB services:

- Site persistence is not supported, because the service referenced by a CNAME can be present at any third-party location.
- Multiple-IP-address response is not supported because one domain cannot have multiple CNAME entries.
- Source IP Hash and Round Robin are the only load balancing methods supported. The Static Proximity method is not supported because a CNAME is not associated with an IP address and static proximity can be maintained only according to the IP addresses.

Note: The **Empty-Down-Response** feature should be enabled on the GSLB virtual server to which you bind the CNAME-based GSLB service. If you enable the **Empty-Down-Response** feature, when a GSLB virtual server is DOWN or disabled, the response to a DNS query, for the domains bound to this virtual server, contains an empty record without any IP addresses, instead of an error code.

To create a CNAME-based GSLB service by using the NetScaler command line

At the NetScaler command prompt, type:

add gslb service <serviceName> **-cnameEntry** <string> **-siteName** <string>

Example

add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -siteName Site-GSLB-East-Coast
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -siteName Site-GSLB-West-Coast

Parameters for creating a CNAME based GSLB service

serviceName (Service Name)

The name of the CNAME-based GSLB service being configured.

cnameEntry (DNS Canonical Name)

The canonical name of the GSLB domain that the GSLB service will handle.

siteName (Site Name)

The name of the GSLB site that the GSLB service represents.

To create a CNAME-based GSLB service by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create GSLB Service dialog box**, set the following parameters:
   - **Service Name***
   - **Site Name***
   - **Type** should be **Canonical name based**.
   - **DNS Canonical name***

   * A required parameter
4. Click **Create**, and then click **Close**.

Changing the GSLB Method

Unlike traditional DNS servers that simply respond with the IP addresses of the configured servers, a NetScaler appliance configured for GSLB responds with the IP addresses of the services, as determined by the configured GSLB method. By default, the GSLB virtual server is set to the least connection method. If all GSLB services are down, the NetScaler responds with the IP addresses of all the configured GSLB services.

GSLB methods are algorithms that the GSLB virtual server uses to select the best-performing GSLB service. After the host name in the Web address is resolved, the client sends traffic directly to the resolved service IP address.

The NetScaler appliance provides the following GSLB methods:

- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

For GSLB methods to work with a remote site, either MEP must be enabled or explicit monitors must be bound to the remote services. If MEP is disabled, RTT, Least Connections, Least Bandwidth, Least Packets and Least Response Time methods default to Round Robin.

The Static Proximity and RTT load balancing methods are specific to GSLB.

Specifying a GSLB Method Other than Static Proximity or Dynamic (RTT)

For information about the Round Robin, Least Connections, Least Response Time, Least Bandwidth, Least Packets, Source IP Hash, or Custom Load method, see .

To change the GSLB method by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver <vServerName> **-lbMethod** GSLBMethod

Example

set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN

To change the GSLB method by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the details pane, select a GSLB virtual server and click **Open**.
3. In the **Configure GSLB Virtual Server** dialog box, on the **Method and Persistence** tab, under **Method**, select a method from the **Choose Method** list.
4. Click **OK**, and verify that the method you selected appears under **Details** at the bottom of the screen.

Configuring Static Proximity

The static proximity method for GSLB uses an IP-address based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The NetScaler appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the NetScaler appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

For the static proximity method to work, you must either configure the NetScaler appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the GSLB method.

Adding a Location File to Create a Static Proximity Database

A static proximity database is a UNIX-based ASCII file. Entries added to this database from a location file are called static entries. Only one location file can be loaded on a NetScaler appliance. Adding a new location file overrides the existing file. The number of entries in the static proximity database is limited by the configured memory in the NetScaler appliance.

The static proximity database can be created in the default format or in a format derived from commercially configured third party databases (such as www.maxmind.com and www.ip2location.com).

These databases vary in the details they provide. There is no strict enforcement of the database file format, except that the default file has format tags. The database files are ASCII files that use a comma as the field delimiter. There are differences in the structure of fields and the representation of IP addresses in the locations.

The format parameter describes the structure of the file to the NetScaler appliance. Specifying an incorrect value for the format option can corrupt the internal data.

Note: The default location of the database file is /var/netscaler/locdb and, in a high availability (HA) setup, an identical copy of the file must be present in the same location on both NetScaler appliances.

The following abbreviations are used in this section:

- **CSHN**. Short name of a country based on the country code standard of ISO-3166.
- **LCN**. Long name of the country.
- **RC**. Region code based on ISO-3166-2 (for US and Canada). The region code “FIPS-10-4” is used for the other regions.

Note: Some databases provide short country names according to ISO-3166 and long country names as well. The NetScaler uses short names when storing and matching qualifiers.

To create a static proximity database, log on to the UNIX shell of the NetScaler appliance and use an editor to create a file with the location details in one of the NetScaler-supported formats.

To add a static location file by using the NetScaler command line

At the NetScaler command prompt, type:

- add locationfile <locationFile> [**-format** <locationFormat>]
- show locationfile [<locationFile>]

Example

```
add locationfile /var/nsmap/locdb/nsgeo1.0 -format netscaler
show locationfile /var/nsmap/locdb/nsgeo1.0
```

Parameters for adding a static location file

locationFile

The name of the location file. Must include the absolute path to the file. If the full path is not given, the default path /var/netscaler/locdb is assumed. In a high availability setup, the static database must be stored in the same location on both systems.

format

The format of the location file. Possible values: netscaler, ip-country, ip-country-isp, ip-country-region-city, ip-country-region-city-isp, geoip-country, geoip-region, geoip-city, geoip-country-org, geoip-country-isp, geoip-city-isp-org. Default: netscaler.

To add a static location file by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Location**.
2. In the details pane, click the **Static Database** tab, and then click **Add**.
3. In the **Create Location File** dialog box, in the **Location Filename** text box, type the name of the location file, or click **Browse** to select the location file (for example, type or select /var/ nsmap/locdb/nsgeo1.0).

   Note: The location file must be existing on the NetScaler appliance.
4. In the **Location Format** box, select the format of the location (for example, netscaler).
5. Click **Create** and click **Close**.

You can view an imported location file database by using the View Database dialog box in the configuration utility. There is no NetScaler command line equivalent.

To view a static location file by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Location**.
2. On the **Static Database** tab, select the location file and then click **View Database**.
3. In the **View Database** dialog box, and click **Find** to use the following controls to filter and sort the database information.
   a. **Search In**. Choose the field to search from the drop-down list.
   b. **Criterion**. Choose the search criterion from the drop-down list. The list contains a standard set of search criteria. "Contains" is the default choice.
   c. **Look For**. Type the text or number to search for.
   d. **Find Now**. Click this button to perform the search.
   e. **Clear**. Click this button to reset the search controls to their initial state.
4. Click **Close** to close the **View Database** dialog box and return to the **Static Database** tab.

To convert a location file into the netscaler format

By default, when you add a location file, it is saved in the netscaler format. You can convert a location file of other formats into the netscaler format. See the list of supported formats in the table, Parameters for adding a static location file.

Note: The nsmap option can be accessed only from the command line interface. The conversion is possible only into the netscaler format.

To convert the static database format, at the NetScaler command prompt, type the following command:

nsmap **-f** <inputFileFormat> **-o** <outputFileName > <inputFileName>

Example

nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.cvs

Adding Custom Entries to a Static Proximity Database

Custom entries take precedence over static entries in the proximity database. You can add a maximum of 50 custom entries. For a custom entry, denote all omitted qualifiers with an asterisk (*) and, if qualifiers have a period or space in the name, enclose the parameter in double quotation marks. The first 31 characters are evaluated for each qualifier.

To add custom entries by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a custom entry to the static proximity database and verify the configuration:

- add location < IPfrom> < IPto> <preferredLocation>
- show location

Example

```
add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
show location
```

Parameters for adding custom entries

IPfrom

Start of the IP address range, in dotted notation.

IPto

End of the IP address range, in dotted notation.

preferredLocation

Qualifiers, in dotted notation, for the IP address range. Maximum length is 198. A qualifier that includes a period (.) or space ( ) must be enclosed in double quotation marks.

To add custom entries by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Location**.
2. On the **Custom Entries** tab, click **Add**.
3. Specify values for the following parameters, which correspond to parameters described in “Parameters for adding custom entries” as shown:
   - **From IP Address***—IPfrom
   - **To IP Address***—IPto
   - **Location Name***—preferredLocation

   * A required parameter
4. Click **Create**, and then click **Close**. The custom entry that you have created appears on the **Custom Entries** tab.

Setting the Location Qualifiers

The database used to implement static proximity contains the location of the GSLB sites. Each location contains an IP address range and up to six qualifiers for that range. The qualifiers are literal strings and are compared in a prescribed order at run time. Every location must have at least one qualifier. The meaning of the qualifiers (context) is defined by the qualifier labels, which are user defined. The NetScaler has two built-in contexts:

Geographic context, which has the following qualifier labels:

- Qualifier 1 – “Continent”
- Qualifier 2 – “Country”
- Qualifier 3 – “State”
- Qualifier 4 – “City”
- Qualifier 5 – “ISP”
- Qualifier 6 – “Organization”

Custom entries, which have the following qualifier labels:

- Qualifier 1 – “Qualifier 1”
- Qualifier 2 – “Qualifier 2”
- Qualifier 3 – “Qualifier 3”
- Qualifier 4 – “Qualifier 4”
- Qualifier 5 – “Qualifier 5”
- Qualifier 6 – “Qualifier 6”

If the geographic context is set with no Continent qualifier, Continent is derived from Country. Even the built-in qualifier labels are based on the context, and the labels can be changed. These qualifier labels specify the locations mapped with the IP addresses used to make static proximity decisions.

To perform a static proximity-based decision, the NetScaler appliance compares the location attributes (qualifiers) derived from the IP address of the local DNS server resolver with the location attributes of the participating sites. If only one site matches, the appliance returns the IP address of that site. If there are multiple matches, the site selected is the result of a round robin on the matching GSLB sites. If there is no match, the site selected is a result of a round robin on all configured sites. A site that does not have any qualifiers is considered a match.

To set the location qualifiers by using the NetScaler command line

At the NetScaler command prompt, type:

set locationparameter **-context** <context> **-q1label** <string> [**-q2label** <string>] [**-q3label** <string>] [**-q4label** <string>] [**-q5label** <string>] [**-q6label** <string>]

Example

set locationparameter -context custom -q1label asia

Parameters for setting the location qualifiers

context

The context in which a static proximity decision is made. Possible Values: geographic, custom.

q1label

The label for the 1st qualifier.

q2label

The label for the 2nd qualifier.

q3label

The label for the 3rd qualifier.

q4label

The label for the 4th qualifier.

q5label

The label for the 5th qualifier.

q6label

The label for the 6th qualifier.

To set the location qualifiers by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Location**.
2. Click **Location Parameters**.
3. In the **Context** drop-down list, select the appropriate context (for example, Custom).
4. In the **Qualifier Label -1** text box, type the qualifier (for example asia).
5. Click **OK**.

Specifying the Proximity Method

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure static proximity and verify the configuration:

- set gslb vserver <vServerName> **-lbMethod** STATICPROXIMITY
- show gslb vserver <vServerName>

Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
```

To specify static proximity by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB Virtual Server that you want to set to static proximity (for example, vserver-GSLB-1).
3. Click **Open.**
4. On the **Method and Persistence** tab, under **Method**, select **Static Proximity** from the **Choose Method** list.
5. Click **OK.**
6. Verify that the **Details** pane shows static proximity as the GSLB method.

Configuring the Dynamic Method (RTT)

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the NetScaler appliance probes the client's local DNS server and gathers RTT metric information. The appliance then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value.

When a client's DNS request for a domain comes to the NetScaler appliance configured as the authoritative DNS for that domain, the appliance uses the RTT value to select the IP address of the best performing site to send it as a response to the DNS request.

The NetScaler appliance uses different mechanisms, such as ICMP echo request / reply (PING), TCP, and UDP to gather the RTT metrics for connections between the local DNS server and participating sites. The appliance first sends a ping probe determine the RTT. If the ping probe fails, a DNS TCP probe is used. If that probe also fails, the appliance uses a DNS UDP probe.

The NetScaler appliance performs UDP probing on port 53 and TCP probing on port 80, and uses the proprietary metrics exchange protocol (MEP) to exchange RTT values between participating sites. After calculating RTT metrics, the appliance sorts the RTT values to identify the data center with the best (smallest) RTT metric.

If RTT information is not available (for example, when a client's local DNS server accesses the site for the first time), the NetScaler appliance selects a site by using the round robin method and directs the client to the site.

To configure the dynamic method, you configure the site's GSLB virtual server for dynamic RTT. You can also set the interval at which local DNS servers are probed to a value other than the default.

Configuring a GSLB Virtual Server for Dynamic RTT

To configure a GSLB virtual server for dynamic RTT, you specify the RTT load balancing method.

The NetScaler appliance regularly validates the timing information for a given local server. If a change in latency exceeds the configured tolerance factor, the appliance updates its database with the new timing information and sends the new value to other GSLB sites by performing a MEP exchange. The default tolerance factor is 5 milliseconds (ms).

The RTT tolerance factor must be the same throughout the GSLB domain. If you change it for a site, you must configure identical RTT tolerance factors on all NetScaler appliances deployed in the GSLB domain.

To configure a GSLB virtual server for dynamic RTT by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver <VserverName> **-lbMethod** RTT **-tolerance** <value>

Example

set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10

Parameters for configuring the dynamic RTT load balancing method

name

The name of the GSLB virtual server for which you are configuring the load balancing method.

lbMethod

The load balancing method being configured for the GSLB virtual server. For the dynamic method, specify RTT.

tolerance

The minimum number of milliseconds by which the RTT metric must change to trigger an update of this metric in the database.

To configure a GSLB virtual server for dynamic RTT by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB Virtual server that you want to set to dynamic RTT (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Method and Persistence** tab, under **Method**, select **Dynamic Method (RTT)** from the **Choose Method** list.
5. To change the tolerance factor, type the new value in the **Tolerance (ms)** text box. (For a description of the tolerance factor, see “Parameters for configuring the dynamic RTT load balancing method.”)
6. Click **OK**.

Setting the Probing Interval of Local DNS Servers

The NetScaler appliance uses different mechanisms, such as ICMP echo request / reply (PING), TCP, and UDP to obtain RTT metrics for connections between the local DNS server and participating GSLB sites. By default, the appliance uses a ping monitor and probes the local DNS server every 5 seconds. The appliance then waits 2 seconds for the response and, if a response is not received in that time, it uses the TCP DNS monitor for probing.

However, you can modify the time interval for probing the local DNS server to accommodate your configuration.

To modify the probing interval by using the NetScaler command line

At the NetScaler command prompt, type:

set lb monitor <monitorName> <type> **-interval** <integer> <units> **-resptimeout** <integer> <units>

Example

set lb monitor monitor-HTTP-1 HTTP -interval 10 sec -resptimeout 5 sec

Parameters for modifying the probing interval

interval

Interval at which probes should be sent.

units

monitor-interval/resptimeout units. Possible values are SEC, MSEC,and MIN. Default value is NSTMUNT_SEC.

type

The type of monitor being configured. The following are valid monitor types:

- **TCP** - The NetScaler appliance establishes a TCP connection with the monitor destination and then closes the connection. If the NetScaler observes TCP traffic to the destination, it does not send TCP monitoring requests. This occurs if LRTM is disabled. By default, LRTM is disabled on this monitor. This is done only for UDP, and the service goes down immediately.

- **TCP-ECV** - The NetScaler appliance establishes a TCP connection with the monitor destination. When the connection is established, the appliance sends specific data to the service by using the -send parameter, and the appliance and expects a specific response through the receive parameter.

- **HTTP** - the NetScaler establishes a TCP connection with the monitor destination. After the connection is established, the NetScaler sends HTTP requests and compares the response code, in the response from the service, with the configured set of response codes.

- **HTTP-ECV** - the NetScaler establishes a TCP connection with the monitor destination. When the connection is established, the NetScaler sends the HTTP data specified by the -send parameter to the service and expects the HTTP response that the -receive parameter specifies. (HTTP body part, not including HTTP headers.) Empty response data matches any response. Expected data may be anywhere in the first 24K bytes of the HTTP body of the response.

- **PING** - the NetScaler sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.

Note: The NetScaler also supports FTP, UDP, DNS, UDP-ECV, TCPS, HTTPS, TCPS-ECV, HTTPS-ECV, LDNS-PING, LDNS-TCP, and LDNS-DNS monitors.

For more information about monitors, see Load Balancing.

resptimeout

Interval after which probe should be marked as FAILED.

To modify the probing interval by using the configuration utility

- In the navigation pane, expand **Load Balancing** and click **Monitors.**
- Select the monitor that you want to modify (for example, ping).
- Click **Open**.
- In the **Configure Monitor** dialog box, on the **Standard Parameters** tab, specify values for the following parameters, which correspond to parameters described in “Parameters for modifying the probing interval” as shown:
  - **Interval**—interval
  - **Response Time-out**—resptimeout (type the interval after which the probe should be marked as FAILED. Specify whether the value represents minutes, milliseconds, or seconds by selecting a value from the adjacent list)
- Click **OK.**

Configuring Persistent Connections

Persistence ensures that a series of client requests for a particular domain name is sent to the same data center instead of being load balanced. If persistence is configured for a particular domain, it takes precedence over the configured GSLB method. Persistence is useful for deployments that deal with e-commerce, such as shopping card usage, where the server needs to maintain the state of the connection to track the transaction. To maintain the state of connection, you must configure persistence on a virtual server. With persistence configured, NetScaler selects a data center to process a client request and forwards the IP address of the selected data center for all subsequent DNS requests. If the configured persistence applies to a site that is down, the NetScaler appliance uses a GSLB method to select a new site, and the new site becomes persistent for subsequent requests from the client.

The GSLB virtual server is responsible for DNS-based site persistence, and it controls the site persistence for a remote GSLB service. The NetScaler appliance supports persistence based on the source IP address or on HTTP cookies.

When you bring a physical service DOWN with a delay time, the physical service goes into the transition out of service (TROFS) state. Site persistence is supported as long as the service is in the TROFS state. That is, if the same client sends a request for the same service within the specified delay time after a service is marked TROFS, the same GSLB site (data center) services the request.

Note: If connection proxy is specified as the site persistence method and if you also want to configure persistence of the physical servers, do not configure SOURCEIP persistence. When the connection is proxied, an IP address owned by the NetScaler is used, and not the actual IP address of the client. Configure methods such as cookie persistence or rule-based persistence on the load balancing virtual server.

Configuring Persistence Based on Source IP Address

With source-IP persistence, when a DNS request is received at a data center, the NetScaler appliance first looks for an entry in the persistence table and, if an entry for the local DNS server exists and the server mentioned in the entry is configured, the IP address of that server is sent as the DNS response.

For the first request from a particular client, the NetScaler appliance selects the best GSLB site for the request and sends its IP address to the client. Since persistence is configured for the source IP address of the client, all subsequent requests by that client or another local DNS server in the same IP subnet are sent the IP address of the GSLB site that was selected for the first request.

For source-IP address based persistence, the same set of persistence identifiers must be configured on the GSLB virtual servers in all data centers. A persistence identifier is a number used by the data centers to identify a particular GSLB virtual server. A cookie transmits the persistence identifier, enabling the NetScaler appliance to identify the domain so that it can forward all appropriate requests to the same domain. When persistence is enabled, the persistence information is also exchanged as part of metrics exchange.

For the NetScaler appliance to support persistence across sites, persistence must be enabled on the GSLB virtual servers of all participating sites. When you use source IP address persistence on the network identifier, you must configure a subnet mask. For any domain, persistence takes precedence over any other configured GSLB method.

To configure persistence based on source IP address by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver <name> **-persistenceType SOURCEIP -persistenceId** <positive_integer> [**-persistMask** <netmask>] -[**timeout** <mins>]

Example

set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -persistenceId 23 -persistMask 255.255.255.255 -

Parameters for configuring persistence based on source IP address

name

The name of the GSLB virtual server for which you are configuring source IP address based persistence.

persistenceType

The type of persistence being configured for the GSLB virtual server. Possible Values: SOURCEIP, None.

persistenceID

A positive integer used to identify the GSLB virtual server on all sites. Minimum value: 1. Maximum value: 65535.

persistMask

The subnet mask used when SOURCEIP based persistence is enabled. Minimum Value: 128.0.0.0. Default: 0xFFFFFFFF.

timeout

The time, in minutes, for which persistence should be in effect for the GSLB virtual server. Minimum value: 2. Maximum value: 1440. Default: 2.

To configure persistence based on source IP address by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Method and Persistence** tab, under Persistence, select SOURCEIP from the Pesistence list and specify values for the following parameters, which correspond to parameters described in “Parameters for configuring persistence based on source IP address” as shown:
   - **Time-out**—timeout
   - **Persistence Id**—persistenceID
   - **IPv4 Netmask or IPv6 Mask length**—persistMask
5. Click **OK**.

Configuring Persistence Based on HTTP Cookies

The NetScaler appliance provides persistence at the HTTP-request level by using connection proxy and HTTP redirect. With these persistence methods, the appliance uses an HTTP cookie (known as a “site cookie”) to reconnect the client to the same server. The NetScaler inserts the site cookie in the first HTTP response.

The site cookie contains information about the selected GSLB service on which the client has a persistent connection. The cookie expiration is based on the cookie timeout configured on the NetScaler appliance. If the virtual server names are not identical on all the sites, you must use the persistence identifier. Cookies inserted are compliant with RFC 2109.

When the NetScaler appliance responds to a client DNS request by sending the IP address of the selected GSLB site, the client sends an HTTP request to that GSLB site. The physical server in that GSLB site adds a site cookie to the HTTP header, and connection persistence is in effect.

If the DNS entry in the client cache expires, and then the client sends another DNS query and is directed to a different GSLB site, the new GSLB site uses the site cookie present in the client request header to implement persistence. If the GSLB configuration at the new site uses connection-proxy persistence, the new site creates a connection to the GSLB site that inserted the site cookie, proxies the client request to the original site, receives a response from the original GSLB site, relays that response back to the client, and closes the connection. If the GSLB configuration uses HTTP redirect persistence, the new site redirects the request to the site that originally inserted the cookie.

Note: Connection proxy persistence can be configured only for local services. However, connection proxy persistence must be enabled for all services in the GSLB configuration.

Connection proxy occurs when the following conditions are satisfied:

- Requests are sent from a domain participating in GSLB. The domain is obtained from the URL/Host header.
- Requests are sent from a local GSLB service whose public IP address matches the public IP address of an active service bound to the GSLB virtual server.
- The local GSLB service has connection proxy enabled.
- The request includes a valid cookie that contains the IP address of an active remote GSLB service.

If one of the conditions is not met, connection proxy does not occur, but a site cookie is added if the local GSLB service has connection proxy enabled AND:

- No site cookie is supplied; OR,

- The site cookie refers to an IP address that is not an active GSLB remote service; OR,
- The cookie refers to the IP address of the virtual server on which the request is received.

The following are the limitations of using connection proxy site cookies:

- Site cookies do not work for non-HTTP(S) protocols.
- If an HTTP request is sent to a back-up virtual server, the virtual server does not add a cookie.
- Site cookies do not work if SSL client authentication is required.
- At the local site, the statistics for a GSLB service on a remote site are not the same as the statistics recorded for that service at the remote site. At the local site, the statistics for a remote GSLB service are slightly higher than the statistics that the remote site records for that same service.

Redirect persistence can be used only:

- For HTTP or HTTPS protocols.
- If the domain name is present in the request (either in the URL or in the HOST header), and the domain is a GSLB domain.
- When the request is received on a backup VIP or a GSLB local service that is in the down state.

To set persistence based on HTTP cookies by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb service <serviceName> **-sitePersistence** (**ConnectionProxy** [-sitePrefix <prefix>] | HTTPredirect **-sitePrefix** <prefix>)

Example

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1
```

Parameters for setting persistence based on HTTP cookies

serviceName

The name of the GSLB service for which connection proxy based cookie persistence is being configured.

sitePersistence

The type of persistence. Possibe Values: connectionProxy, HTTPredirect, None.

sitePrefix

This is a mandatory parameter when you configure HTTP redirect based persistence on a GSLB service. When the service is bound to a GSLB virtual server, for each bound service-domain pair, a GSLB site domain is generated internally by concatenating the service's siteprefix and the domain's name. If a special string "NONE" is specified, the siteprefix string is not set.

To set persistence based on cookies by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Services**.
2. In the **GSLB Services** pane, select the service that you want to configure for site persistence (for example, service-GSLB-1).
3. Click **Open**.
4. On the **Advanced** tab, under **Site Persistence type**, specify values for the following paramters, which correspond to paramters described in “Parameters for setting persistence based on HTTP cookies” as shown:
   - **Site Persistence type**—sitePersistence
   - **Site Prefix**—sitePrefix
5. Click **OK**.

Configuring Transition Out-Of-Service State (TROFS) in GSLB

When you configure persistence on a GSLB virtual server to which a service is bound, the service continues to serve requests from the client even after it is disabled, accepting new requests or connections only to honor persistence. After a configured period of time, known as the graceful shutdown period, no new requests or connections are directed to the service, and all of the existing connections are closed.

When disabling a service, you can specify a graceful shutdown period, in seconds, by using the delay argument. During the graceful shutdown period, if the service is bound to a virtual server, its state appears as Out of Service.

Configuring Dynamic Weights for Services

In a typical network, there are servers that have a higher capacity for traffic than others. However, with a regular load balancing configuration, the load is evenly distributed across all services even though different services represent servers with different capacities.

To optimize your GSLB resources, you can configure dynamic weights on a GSLB virtual server. The dynamic weights can be based on either the total number of services bound to the virtual server or the sum of the weights of the individual services bound to the virtual server. Traffic distribution is then based on the weights configured for the services.

When dynamic weights are configured on the GSLB virtual server, requests are distributed according to the load balancing method, the weight of the GSLB service, and the dynamic weight. The product of the weight of the GSLB service and the dynamic weight is known as the cumulative weight. Therefore, when dynamic weight is configured on the GSLB virtual server, requests are distributed on the basis of the load balancing method and the cumulative weight.

When dynamic weight for a virtual server is disabled, the numerical value is set to 1. This ensures that the cumulative weight is a non-zero integer at all times.

Dynamic weight can be based on the total number of active services bound to load balancing virtual servers or on the weights assigned to the services.

Consider a configuration with two GSLB sites configured for a domain and each site has two services that can serve the client. If a service at either site goes down, the other server in that site has to handle twice as much traffic as a service at the other site. If dynamic weight is based on the number of active services, the site with both services active has twice the weight of the site with one service down and therefore receives twice as much traffic.

Alternatively, consider a configuration in which the services at the first site represent servers that are twice as powerful as servers at the second site. If dynamic weight is based on the weights assigned to the services, twice as much traffic can be sent to the first site as to the second.

Note: For details on assigning weights to load balancing services, see Assigning Weights to Services.

As an illustration of how dynamic weight is calculated, consider a GSLB virtual server that has a GSLB service bound to it. The GSLB service represents a load balancing virtual server that in turn has two services bound to it. The weight assigned to the GSLB service is 3. The weights assigned to the two services are 1 and 2 respectively. In this example, when dynamic weight is set to:

- **Disabled**:The cumulative weight of the GSLB virtual server is the product of the dynamic weight (disabled = 1) and the weight of the GSLB service (3), so the cumulative weight is 3.

- **SERVICECOUNT**: The dynamic weight is the sum of the number of services bound to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.
- **SERVICECOUNT**: The dynamic weight is the sum of the number of services bound to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.

Note: Dynamic weights are not applicable when content switching virtual servers are configured.

To configure a GSLB virtual server to use dynamic weights by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver <name> **-dynamicWeight** SERVICECOUNT | SERVICEWEIGHT

Example

set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT

To set GSLB virtual server to use dynamic weights by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB virtual server for which you want to set dynamic weights (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Method and Persistence** tab, under **Method**, select **SERVICECOUNT** or **SERVICEWEIGHT** from the **Dynamic Weight** list.
5. Click **OK**.

Monitoring GSLB Services

When you bind a remote service to a GSLB virtual server, the GSLB sites exchange metric information, including network metric Information, which is the round-trip-time and persistence Information.

If a metric exchange connection is momentarily lost between any of the participating sites, the remote site is marked as DOWN and load balancing is performed on the remaining sites that are UP. When metric exchange for a site is DOWN, the remote services belonging to the site are marked DOWN as well.

The NetScaler appliance periodically evaluates the state of the remote GSLB services by using either MEP or monitors that are explicitly bound to the remote services. Binding explicit monitors to local services is not required, because the state of the local GSLB service is updated by default using the MEP. However, you can bind explicit monitors to a remote service. When monitors are explicitly bound, the state of the remote service is not controlled by the metric exchange.

By default, when you bind a monitor to a remote GSLB service, the NetScaler appliance uses the state of the service reported by the monitor. However, you can configure the NetScaler appliance to use monitors to evaluate services in the following situations:

- Always use monitors (default setting).
- Use monitors when MEP is DOWN.
- Use monitors when remote services and MEP are DOWN.

The second and third of the above settings enable the NetScaler to stop monitoring when MEP is UP. For example, in a hierarchical GSLB setup, a GSLB site provides the MEP information about its child sites to its parent site. Such an intermediate site may evaluate the state of the child site as DOWN because of network issues, though the actual state of the site is UP. In this case, you can bind monitors to the services of the parent site and disable MEP to determine the actual state of the remote service. This option enables you to control the manner in which the states of the remote services are determined.

To use monitors, first create them, and then bind them to GSLB services.

Adding or Removing Monitors

To add a monitor, you specify the type and the port. You cannot remove a monitor that is bound to a service. You must first unbind the monitor from the service.

To add a monitor by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a monitor and verify the configuration:

- add lb monitor <name> **-type** <monitorType> **-destPort** <portNumber>
- show lb monitor <name>

Example

add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1

To remove a monitor by using the NetScaler command line

At the NetScaler command prompt, type:

rm lb monitor <name>

Parameters for adding a monitor

name

The name of the monitor being created. This alphanumeric string is required and cannot be changed after the monitor is created. The name must not exceed 31 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

type

The type of monitor being configured. The following are valid monitor types:

- **TCP** - The NetScaler appliance establishes a TCP connection with the monitor destination and then closes the connection. If the NetScaler observes TCP traffic to the destination, it does not send TCP monitoring requests. This occurs if LRTM is disabled. By default, LRTM is disabled on this monitor. This is done only for UDP, and the service goes down immediately.
- **TCP-ECV** - The NetScaler appliance establishes a TCP connection with the monitor destination. When the connection is established, the appliance sends specific data to the service by using the -send parameter, and the appliance and expects a specific response through the -receive parameter.
- **HTTP** - the NetScaler establishes a TCP connection with the monitor destination. After the connection is established, the NetScaler sends HTTP requests and compares the response code, in the response from the service, with the configured set of response codes.
- **HTTP-ECV** - the NetScaler establishes a TCP connection with the monitor destination. When the connection is established, the NetScaler sends the HTTP data specified by the -send parameter to the service and expects the HTTP response that the -receive parameter specifies. (HTTP body part, not including HTTP headers.) Empty response data matches any response. Expected data may be anywhere in the first 24K bytes of the HTTP body of the response.
- **PING** - the NetScaler sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.

Note: The NetScaler also supports FTP, UDP, DNS, UDP-ECV, TCPS, HTTPS, TCPS-ECV, HTTPS-ECV, LDNS-PING, LDNS-TCP, and LDNS-DNS monitors.

For more information about monitors, see Load Balancing.

destPort

Destination TCP/UDP port of the probe (the port of the dispatcher to which the probe is sent). The port can be different from the server port to which the monitor is bound. The value 0 (zero) directs the probes to the bound server's port. This parameter has no effect on PING type monitors.

To add a monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for adding a monitor” as shown:
   - **Name***—name
   - **Type***—type

   * A required parameter
4. On the **Standard Parameters** tab, in the **Destination Port** text box, type the destination port number (see “destPort” in the above parameter list).
5. Click **Create**, and then click **Close.**

Binding Monitors to a GSLB Service

Once you create monitors, you must bind them to GSLB services. When binding monitors to the services, you can specify a weight for the monitor. After binding one or more weighted monitors, you can configure a monitor threshold for the service. This threshold takes the service down if the sum of the bound monitor weights falls below the threshold value.

Note: In the configuration utility, you can set both the weight and the monitoring threshold at the same time that you bind the monitor. When using the command line, you must issue a separate command to set the service's monitoring threshold.

To bind the monitor to the GSLB service by using the NetScaler command line

At the NetScaler command prompt, type:

bind monitor <name> <serviceName> [**-state** (**Enabled** | **Disabled**)] **-weight** <positiveInteger>

Example

bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2

To set the monitoring threshold for a GSLB service by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb service <ServiceName> **-monThreshold** <PositiveInteger>

Example

set gslb service service-GSLB-1 -monThreshold 9

Parameters for binding a monitor to a GSLB service

name

The name of the monitor to be bound to the service.

serviceName

The name of the service to which to bind the monitor.

weight

The weight to assign to the service. Minimum value: 1. Maximum value: 100. Default: 1.

monThreshold

The monitoring threshold for the service. Minimum value: 0. Maximum value: 65535.

To bind the monitor to the GSLB service by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Services**.
2. The details pane, select the service to which you want to bind the monitor (for example, select service-GSLB-1).
3. Click **Open**.
4. In the **Configure GSLB Service** dialog box, on the **Monitors** tab, select the monitor that you want to bind to the service (for example, monitor-HTTP-1).
5. Click **Add**.
6. In the **Configured** table, you can select the newly assigned monitor and enter a new weight value.
7. To enable the monitor, make sure the **State** check box is selected.
8. Repeat the preceding steps to add additional monitors.
9. In the **Monitor Threshold** text box, you can enter a threshold value.
10. Click **OK**.

Monitoring GSLB Sites

The NetScaler appliance uses MEP or monitors to determine the state of the GSLB sites. You can configure a GSLB site to always use monitors (the default), use monitors when MEP is down, or use monitors when both the remote service and MEP are down. In the latter two cases, the NetScaler appliance stops monitoring when MEP returns to the UP state.

To configure monitor triggering by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb site <name> **-triggerMonitor** (ALWAYS | MEPDOWN | MEPDOWN_SVCDOWN)

Example

```
> set gslb site Site-GSLB-North-America –triggerMonitor Always
 Done
```

To configure monitor triggering by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Sites**.
2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, in the **Trigger Monitors** drop-down list, select an option for when to trigger monitoring.
4. Click **OK**.

Protecting the GSLB Setup against Failure

You can protect your GSLB setup against failure of a GSLB site or a GSLB virtual server by configuring a backup GSLB virtual server, configuring the NetScaler appliance to respond with multiple IP addresses, or configuring a Backup IP address for a GSLB domain. You can also divert excess traffic to a backup virtual server by using spillover.

Configuring a Backup GSLB Virtual Server

Configuring a backup entity for a GSLB virtual server ensures that DNS traffic to a site is not interrupted if the GSLB virtual server goes down. The backup entity can be another GSLB virtual server, or it can be a backup IP address. With a backup entity configured, if the primary GSLB virtual server goes down, the backup entity handles DNS requests. To specify what should happen when the primary GSLB virtual server comes back up again, you can configure the backup entity to continue handling traffic until you manually enable the primary virtual server to take over (using the disablePrimaryOnDown option), or you can configure a timeout period after which the primary takes over.

If you configure both the timeout and the disablePrimaryOnDown option for the backup entity, the backup session time-out takes precedence over the disablePrimaryOnDown setting.

To configure a backup GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a GSLB virtual server as a backup virtual server and verify the configuration:

- set gslb vserver <name> **-backupVServer** <name> [**-backupSessionTimeout** <timeoutValue>] [**-disablePrimaryOnDown** (ENABLED | DISABLED)]
- show gslb vserver <vserverName>

Example

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -backupSessionTimeout 3 -disablePrimaryOnD
show gslb vserver vserver-GSLB-1
```

Parameters for configuring a backup GSLB virtual server

name

The name of the GSLB virtual server for which you are configuring a backup.

backupVServer

The name of the GSLB virtual server being configured as a backup.

backupSessionTimeout

The time, in minutes, after which the former primary GSLB virtual becomes primary again after returning to the UP state.

disablePrimaryOnDown

Require manual intervention to return the former primary GSLB virtual server to primary status.

To set GSLB virtual server as a backup virtual server by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Advanced** tab, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a backup GSLB virtual server” as shown:
   - **Backup VServer**—backupVServer
   - **Backup Session Time-out (mins)**—backupSessionTimeout
   - **Disable Primary When Down**—disablePrimaryOnDown
5. Click **OK**.

Configuring a GSLB Setup to Respond with Multiple IP Addresses

A typical DNS response contains the IP address of the best performing GSLB service. However, if you enable multiple IP response (MIR), the NetScaler appliance sends the best GSLB service as the first record in the response and adds the remaining active services as additional records. If MIR is disabled (the default), the NetScaler appliance sends the best service as the only record in the response.

To configure a GSLB virtual server for multiple IP responses by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a GSLB virtual server for multiple IP responses and verify the configuration:

- set gslb vserver<vserverName> **-MIR** (ENABLED | DISABLED)
- show gslb vserver <vserverName>

Example

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
 show gslb vserver <vserverName>
```

To set a GSLB virtual server for multiple IP responses by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Advanced** tab, under **When this VServer is "UP,"** select the **Send all "active" service IP in response (MIR)** check box.
5. Click **OK**.

Configuring a GSLB Virtual Server to Respond with an Empty Address Record When DOWN

A DNS response can contain either the IP address of the requested domain or an answer stating that the IP address for the domain is not known by the DNS server, in which case the query is forwarded to another name server. These are the only possible responses to a DNS query.

When a GSLB virtual server is disabled or in a DOWN state, the response to a DNS query for the GSLB domain bound to that virtual server contains the IP addresses of all the services bound to the virtual server. However, you can configure the GSLB virtual server to in this case send an empty down response (EDR). When this option is set, a DNS response from a GSLB virtual server that is in a DOWN state does not contain IP address records, but the response code is successful.This prevents clients from attempting to connect to GSLB sites that are down.

Note: You must configure this setting for each virtual server to which you want it to apply.

To configure a GSLB virtual server for empty down responses by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver<vserverName> **-EDR** (ENABLED | DISABLED)

Example

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
 Done
```

To set a GSLB virtual server for empty down responses by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Advanced** tab, under **When this VServer is "Down,"** select the **Do not send any service's IP address in response (EDR)** check box.
5. Click **OK**.

Configuring a Backup IP Address for a GSLB Domain

You can configure a backup site for your GSLB configuration. With this configuration in place, if all of the primary sites go DOWN, the IP address of the backup site is provided in the DNS response.

Typically, if a GSLB virtual server is active, that virtual server sends a DNS response with one of the active site IP addresses as selected by the configured GSLB method. If all the configured primary sites in the GSLB virtual server are inactive (in the DOWN state), the authoritative domain name system (ADNS) server or DNS server sends a DNS response with the backup site's IP address.

Note: When a backup IP address is sent, persistence is not honored.

To set a backup IP address for a domain by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set a backup IP address and verify the configuration:

- set gslb vserver <vserverName> **-domainName** www.abc.com **-backupIP** <IPAddress>
- show gslb vserver <vserverName>

Example

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP 10.102.29.66
show gslb vserver vserver-GSLB-1
```

Parameters for configuring a back up IP address for a domain

vserverName

The name of the GSLB virtual server to which the domain you are configuring a back up IP address for is bound.

domainName

The name of the domain for which a back up IP address is being configured.

backupIP

The IP address of the backup service. This IP address is used when all services bound to the domain are down, or when the backup chain is down.

To set a backup IP address for a domain by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. In the **GSLB Virtual Servers** pane, select the GSLB virtual server to which you want to bind the backup domain (for example, vserver-GSLB-1).
3. Click **Open**.
4. On the **Domains** tab, select a domain and click **Open**.
5. In the **Configure GSLB Domain** dialog box, in the **Backup IP** text box, type the IP address of the backup domain.
6. Click **OK**.

Diverting Excess Traffic to a Backup Virtual Server

Once the number of connections to a primary GSLB virtual server exceeds the configured threshold value, you can use the spillover option to divert new connections to a backup GSLB virtual server. This threshold value can be calculated dynamically or set manually. Once the number of connections to the primary virtual server drops below the threshold, the primary GSLB virtual server resumes serving client requests.

You can configure persistence with spillover. When persistence is configured, new clients are diverted to the backup virtual server if that client is not already connected to a primary virtual server. When persistence is configured, connections that were diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections to the primary virtual server drops below the threshold. Instead, the backup virtual server continues to process those connections until they are terminated by the user. Meanwhile, the primary virtual server accepts new clients.

The threshold can be measured either by the number of connections or by the bandwidth.

If the backup virtual server reaches the configured threshold and is unable to take any additional load, the primary virtual server diverts all requests to the designated redirect URL. If a redirect URL is not configured on the primary virtual server, subsequent requests are dropped.

The spillover feature prevents the remote backup GSLB service (backup GSLB site) from getting flooded with client requests when the primary GSLB virtual server fails. This occurs when a monitor is bound to a remote GSLB service, and the service experiences a failure that causes its state to go DOWN. The monitor continues to keep the state of the remote GSLB service UP, however, because of the spillover feature.

As part of the resolution to this problem, two states are maintained for a GSLB service, the primary state and effective state. The primary state is the state of the primary virtual server and the effective state is the cumulative state of the virtual servers (primary and backup chain). The effective state is set to UP if any of the virtual servers in the chain of virtual servers is UP. A flag that indicates that the primary VIP has reached the threshold is also provided. The threshold can be measured by either the number of connections or the bandwidth.

A service is considered for GSLB only if its primary state is UP. Traffic is directed to the backup GSLB service only when all the primary virtual servers are DOWN. Typically, such deployments will have only one backup GSLB service.

Adding primary and effective states to a GSLB service has the following effects:

- When source IP persistence is configured, the local DNS is directed to the previously selected site only if the primary virtual server on the selected site is UP and below threshold. Persistence can be ignored in the round robin mode.

- If cookie-based persistence is configured, client requests are redirected only when the primary virtual server on the selected site is UP.
- If the primary virtual server has reached its saturation and the backup VIP(s) is absent or down, the effective state is set to DOWN.
- If external monitors are bound to an HTTP-HTTPS virtual server, the monitor decides the primary state.
- If there is no backup virtual server to the primary virtual server and the primary virtual server has reached its threshold, the effective state is set to DOWN.

To configure a backup GSLB virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a backup GSLB virtual server and verify the configuration:

- set gslb vserver <name> **-soMethod** <method> **-soThreshold** <threshold> **-soPersistence** (**ENABLED** | **DISABLED**) **-soPersistenceTimeout** <timeout>
- show gslb vserver <name>

Example

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000 -soPersistence ENABLED -soPersis
show gslb vserver Vserver-GSLB-1
```

Parameters for configuring a backup GSLB virtual server

name

The name of the GSLB virtual server for which a backup virtual server is being configured.

soMethod

The type of spillover used to divert traffic to the backup GSLB virtual server when the primary virtual server reaches the threshold. Possible values:

- **CONNECTION**. Spillover based on number of connections exceeding the threshold.
- **DYNAMICCONNECTION**. Spillover based on the combined number of connections exceeding the threshold.

- **BANDWIDTH**. Spillover based on combined incoming and outgoing bandwidth.
- **HEALTH**. Spillover occurs if bound and active services and service groups fall below a threshold relative to all bound elements.
- **NONE**.

soThreshold

The threshold value that decides when traffic must spill over to the back up virtual server. The following threshold values are supported:

- For the **CONNECTION** (or) **DYNAMICCONNECTION** spillover type, the threshold value is the maximum number of connections that the sites under the primary GSLB virtual server will handle before spillover occurs.
- For the **BANDWIDTH** spillover type, the threshold value is the amount of incoming and outgoing traffic (in kilobits per second) that the GSLB virtual server will handle before spillover occurs. Minimum value: 1. Maximum value: 4,294,967,294.
- For **HEALTH**, the threshold value is a positive integer from 1 through 99. This integer represents a percentage of the sum of the binding weights of all of the enabled, bound, and active GSLB services and service groups relative to the sum of the binding weights of all enabled and bound services and service groups (active and inactive).

soPersistence

The configured spillover persistence state. If you enable spillover persistence, the NetScaler appliance maintains source-IP based persistence over the primary virtual server and backup virtual servers. Possible values: ENABLED, DISABLED. Default: DISABLED.

soPersistenceTimeout

The configured time-out value, in minutes, for spillover persistence. Minimum value: 2. Maximum value: 1440. Default: 2.

backupVServer

The name of the GSLB virtual server being configured as a backup.

backupSessionTimeout

The time, in minutes, after which the former primary GSLB virtual becomes primary again after returning to the UP state.

To configure a backup GSLB virtual server by using the configuration utility

1. In the navigation pane, expand **GSLB**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure as a backup (for example, Vserver-LB-1), and then click **Open**.
3. On the **Advanced** tab, under **Spillover**, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a backup GSLB virtual server" as shown:
   - **Method**— soMethod
   - **Threshold**— soThreshold
   - **Persistence Time-out (min)** — soPersistenceTimeout
4. Select the **Persistence** check box.
5. Click **OK**.

Managing Client connections

To facilitate management of client connections, you can enable delayed cleanup of connections to the virtual server. You can then manage local DNS traffic by configuring DNS policies.

Enabling Delayed Cleanup of Virtual Server Connections

The state of a virtual server depends on the states of the services bound to it, and the state of each service depends on the monitors bound to it. If a server is slow or down, the monitoring probes time out and the service that represents the server is marked as DOWN. A virtual server is marked as DOWN only when all services bound to it are marked as DOWN. You can configure services and virtual servers to either terminate all connections when they go down, or allow the connections to go through. The latter setting is for situations in which a service is marked as DOWN because of a slow server.

When you configure the down state flush option, the NetScaler appliance performs a delayed cleanup of connections to a GSLB service that is down.

To enable delayed cleanup of virtual server connections by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure delayed connection cleanup and verify the configuration:

- set glsb service <name> **-downStateFlush** (ENABLED | DISABLED)
- show gslb service <name>

Example

```
> set gslb service Service-GSLB-1 -downStateFlush ENABLED
 Done
> show gslb service Service-GSLB-1
 Done
```

Parameters for delayed connection cleanup

name

The name of the GSLB service for which delayed connection cleanup is being configured.

downStateFlush

Enables or disables delayed cleanup of connections to the GSLB service. Possible Values: ENABLED or DISABLED.

To enable delayed cleanup of virtual server connections by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Services**.
2. In the **GSLB Services** pane, select the service (for example, service-GSLB-1), and then click **Open**.
3. On the **Advanced tab**, select the **Down state flush** check box.
4. Click **OK**.

Managing Local DNS Traffic by Using DNS Policies

You can use DNS policies to implement site affinity by directing traffic from the IP address of a local DNS resolver or network to a predefined target GSLB site. This is configured by creating DNS policies with DNS expressions and binding the policies globally on the NetScaler appliance.

DNS Expressions

The NetScaler appliance provides certain predefined DNS expressions that can be used for configuring actions specific to a domain. Such actions can, for example, drop certain requests, select a specific view for a specific domain, or redirect certain requests to a specific location.

These DNS expressions (also called *rules*) are combined to create DNS policies that are then bound globally on the NetScaler appliance.

Following is the list of predefined DNS qualifiers available on the NetScaler appliance:

- `CLIENT.UDP.DNS.DOMAIN.EQ("domainname")`
- `CLIENT.UDP.DNS.IS_AREC`
- `CLIENT.UDP.DNS.IS_AAAAREC`
- `CLIENT.UDP.DNS.IS_SRVREC`
- `CLIENT.UDP.DNS.IS_MXREC`
- `CLIENT.UDP.DNS.IS_SOAREC`
- `CLIENT.UDP.DNS.IS_PTRREC`
- `CLIENT.UDP.DNS.IS_CNAME`
- `CLIENT.UDP.DNS.IS_NSREC`
- `CLIENT.UDP.DNS.IS_ANYREC`

The `CLIENT.UDP.DNS.DOMAIN` DNS expression can be used with string expressions. If you are using domain names as part of the expression, they must end with a period (.). For example, `CLIENT.UDP.DNS.DOMAIN.ENDSWITH("abc.com.")`

To create an expression by using the configuration utility

1. Click the icon next to the **Expression** text box. Click **Add**. (Leave the Flow Type and Protocol drop-down list boxes empty.) Follow these steps to create a rule.
2. In the **Qualifier** box, select a qualifier (for example, LOCATION).
3. In the **Operator** box, select an operator (for example, ==).
4. In the **Value** box, type a value (for example, Asia.Japan....).
5. Click **OK**. Click **Create** and click **Close**. The rule is created.
6. Click **OK**.

Configuring DNS Policies

DNS policies operate on a location database that uses static and custom IP addresses. The attributes of the incoming local DNS request are defined as part of an expression, and the target site is defined as part of a DNS policy. While defining actions and expressions, you can use a pair of single quotation marks (‘’) as a wildcard qualifier to specify more than one location. When a DNS policy is configured and a GSLB request is received, the custom IP address database is first queried for an entry that defines the location attributes for the source:

- When a DNS query comes from an LDNS, the characteristics of the LDNS are evaluated against the configured policies. If they match, an appropriate action (site affinity) is executed. If the LDNS characteristics match more than one site, the request is load balanced between the sites that match the LDNS characteristics.
- If the entry is not found in the custom database, the static IP address database is queried for an entry, and if there is a match, the above policy evaluation is repeated.
- If the entry is not found in either the custom or static databases, the best site is selected and sent in the DNS response on the basis of the configured load balancing method.

The following restrictions apply to DNS policies created on the NetScaler appliance.

A maximum of 64 policies are supported.

- DNS policies are global to the NetScaler and cannot be applied to a specific virtual server or domain.
- Domain or virtual server specific binding of policy is not supported.

You can use DNS policies to direct clients that match a certain IP address range to a specific site. For example, if you have a GSLB setup with multiple GSLB sites that are separated geographically, you can direct all clients whose IP address is within a specific range to a particular data center.

To add a DNS policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a DNS policy and verify the configuration:

- add dns policy <name> <rule> **-viewName** <string>
- show dns policy <name>

Example

```
add dns policy policy-GSLB-1 “CLIENT.UDP.DNS.DOMAIN.EQ(\“domainname\”)” -view private
show dns policy policy-GSLB-1
```

To remove a configured DNS policy by using the NetScaler command line

At the NetScaler command prompt, type:

rm dns policy <name>

Parameters for configuring a DNS policy

name

The name of the DNS policy being created.

rule

Expression to be used by the dns policy.

viewName

The name of the DNS view to be associated with the DNS policy.

For details on DNS views, see Adding DNS Views.

To add a DNS policy by using the configuration utility

1. In the navigation pane, expand **DNS** and click **Policies**.
2. In the details pane, click **Add**.
3. In the **Policy Name** box, type a name for the DNS policy (for example, policy-GSLB-1).
4. Select **View Name** radio button and, select the view in the **View Name** drop down list, or click **New** to create a view.
5. Under **Expression**, click **Add…** and in the **Add Expression** dialog box, do the following:
    a. In the first drop-down box, select **CLIENT**.
    b. In the second drop-down list box, select **UDP**.
    c. In the next drop-down list box, select **DNS**.
    d. In the next drop-down list box, select **DOMAIN**.
    e. In the next drop-down list box, select **EQ(String)**.
    f. In the next text box, type the domain name (for example, abc.com).
6. Click **OK** and click **Close**. The expression is displayed under Expression in the **Create DNS Policy** dialog box.
7. Click **Create**, and then click **Close**.

Binding DNS Policies

DNS policies are bound globally on the NetScaler appliance and are available for all configured GSLB virtual servers. Even though DNS policies are globally bound, policy execution can be limited to a specific GSLB virtual server by specifying the domain in the expression.

To bind a DNS policy globally by using the NetScaler command line

At the NetScaler command prompt, type:

bind dns global <name> PriorityValue

Example

bind dns global policy-GSLB-1 10

To bind a DNS policy globally by using the configuration utility

1. In the navigation pane, expand **DNS** and click **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **Bind/Unbind DNS Policy(s) to Global** dialog box, click **Insert Policy**.
4. In the **Policy Name** column, select the policy, from the list, that you want to bind.
5. Click **OK**.

To view the global bindings of a DNS policy by using the NetScaler command line

At the NetScaler command prompt, type:

show dns global

To view the global bindings of a DNS policy by using the configuration utility

1. In the navigation pane, expand **DNS** and click **Policies**.
2. In the details pane, click **Global Bindings**. The global bindings of all DNS policies appear in this dialog box.

Adding DNS Views

You can configure DNS views to identify various types of clients and provide an appropriate IP address to a group of clients who query for the same GSLB domain. DNS views are configured by using DNS policies that select the IP addresses sent back to the client.

For example, if you have configured GSLB for your company's domain and have the server hosted in your company's network, clients querying for the domain from within your company's internal network can be provided with the server's internal IP address instead of the public IP address. Clients that query DNS for the domain from the Internet, on the other hand, can be provided the domain's public IP address.

To add a DNS view, you assign it a name of up to 31 characters. The leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ). After adding the view, you configure a policy to associate it with clients and a part of the network, and you bind the policy globally. To configure and bind a DNS policy, see Configuring DNS Policies and Binding DNS Policies.

To add a DNS view by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a DNS view and verify the configuration:

- add dns view <viewName>
- show dns view <viewName>

Example

```
add dns view PrivateSubnet
show dns view PrivateSubnet
```

To remove a DNS view by using the NetScaler command line

At the NetScaler command prompt, type:

rm dns view <viewName>

To add a DNS view by using the configuration utility

1. In the navigation pane, expand **DNS** and click **Views**.
2. In the details pane, click **Add**.
3. In the **Create DNS view** dialog box, in the **Name** text box, enter the name of the DNS view.
4. Click **Create**, and then click **Close**. The DNS view that you created appears in the **Views** pane.

For details on how to create a DNS policy, see Configuring DNS Policies and for details on how to bind DNS policies globally, see Binding DNS Policies.

Configuring GSLB for Commonly Used Deployment Scenarios

GSLB is commonly used in the following deployment scenarios:

- GSLB for disaster recovery
- GSLB based on proximity
- GSLB based on scalability
- GSLB based on the number of Access Gateway users
- GSLB for XenDesktop

Configuring GSLB for Disaster Recovery

Disaster recovery capability is critical, because downtime is costly. A NetScaler appliance configured for GSLB forwards traffic to the least-loaded or the best-performing data center. This configuration, referred to as an active-active setup, not only improves performance, but also provides immediate disaster recovery by routing traffic to other data centers if a data center that is part of the setup goes down. Alternatively, you can configure an active-standby GSLB setup for disaster recovery only.

Configuring GSLB for Disaster Recovery in an Active-Standby Data Center Setup

A conventional disaster recovery setup includes an active data center and a standby data center. The standby data center is a remote site. When a failover occurs as a result of a disaster event that causes the primary active data center to be inactive, the standby data center becomes operational.

Configuring disaster recovery in an active-standby data-center setup consists of the following tasks.

- Create the active data center.
    - Add a local GSLB site.
    - Add a GSLB vserver, which represents the active data center.
    - Bind the domain to the GSLB virtual server.
    - Add gslb services and bind the services to active GSLB virtual server.
- Create the standby data center.
    - Add a remote gslb site.
    - Add a gslb vserver, which represents standby data center.
    - Add gslb services which represents standby data center and bind the services to the standby gslb vserver.
    - Designate the standby data center by configuring the standy GSLB virtual server as the backup virtual server for the active GSLB virtual server.

Once you have configured the primary data center, replicate the configuration for the backup data center and designate it as the standby GSLB site by designating a GSLB virtual server at that site as the backup virtual server.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

To designate the standby GSLB site by using the NetScaler command line

At both the active site and the remote site, at the NetScaler command prompt, type:

set gslb vserver <VserverName> **-backupVserver** <ServerName>

Example

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
```

To configure the standby site by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. Select the **GSLB virtual server for the primary site** and click **Open**.
3. In the **Configure GSLB Virtual Server** dialog box, on the **Advanced** tab, in the **Backup VServer** drop-down list box, select a backup virtual server.
4. Click **OK**.

By default, once the primary virtual server becomes active, it starts receiving traffic. However, if you want the traffic to be directed to the backup virtual server even after the primary virtual server becomes active, use the **'disable primary on down'** option.

Configuring for Disaster Recovery in an Active-Active Data Center Setup

An active-active GSLB deployment, in which both GSLB sites are active, removes any risk that may arise in having a standby data center. With such a setup, web or application content can be mirrored in geographically separate locations. This ensures that data is consistently available at each distributed data center.

To configure GSLB for disaster recovery in an active-active data center set up, you must first configure the basic GSLB setup on the first data center and then configure all other data centers.

First create at least two GSLB sites. Then, for the local site, create GSLB a virtual server and GSLB services and bind the services to the virtual servers. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server in the local site. Finally, at the local site, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Once you have configured the first data center, replicate the configuration for other data centers part of the setup.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Configuring for Disaster Recovery with Weighted Round Robin

When you configure GSLB to use the weighted round robin method, weights are added to the GSLB services and the configured percentage of incoming traffic is sent to each GSLB site. For example, you can configure your GSLB setup to forward 80 percent of the traffic to one site and 20 percent of the traffic to another. After you do this, the NetScaler appliance will send four requests to the first site for each request that it sends to the second.

To set up the weighted round robin method, first create two GSLB sites, local and remote. Next, for the local site create a GSLB virtual server and GSLB services, and bind the services to the virtual servers. Configure the GSLB method as round robin. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Each service that represents a physical server in the network has weights associated with it. Therefore the GSLB service is assigned a dynamic weight that is the sum of weights of all services bound to it. Traffic is then split between the GSLB services based on the ratio of the dynamic weight of the particular service to the total weight. You can also configure individual weights for each GSLB service instead of the dynamic weight.

If the services do not have weights associated with them, you can configure the GSLB virtual server to use the number of services bound to it to calculate the weight dynamically.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you configure a basic GSLB setup, you must configure the weighted round robin method such that the traffic is split between the configured GSLB sites according to the weights configured for the individual services.

To configure a virtual server to assign weights to services by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands, depending upon whether you want to create a new load balancing virtual server or configure an existing one:

- add lb vserver <LBVserverName> **-weight** <WeightValue> <ServiceName>
- set lb vserver <LBVserverName> **-weight** <WeightValue> <ServiceName>

Example

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
```

To set dynamic weight by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver <GSLBVserverName> **-dynamicWeight** DynamicWeightType

Example

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
```

To add weights to the GSLB services by using the NetScaler command line

At the NetScaler command prompt, type:

set gslb vserver <GSLBVserverName> **-serviceName** GSLBServiceName **-weight** WeightValue

Example

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
```

Parameters for configuring a backup GSLB virtual server

lbVserverName

The name of the load balancing virtual server whose services you are configuring weights for.

serviceName

The name of the service whose weights you are configuring.

weight

The weight associated with the service. Minimum Value: 1, Maximum Value: 100.

dynamicWeight

Configures the GSLB virtual server to use either the service count or the cumulative service weights as its dynamic weight. Possible Values: SERVICECOUNT, SERVICEWEIGHT, DISABLED Default Value: DISABLED.

To configure a virtual server to assign weights to services by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
2. Select the virtual server (for example, Vserver-LB-1) and click **Open**.
3. On the Services tab, in the **Weights** spin box, type or select the weight of a service (for example, 4) next to Service-HTTP-1).
4. Click **OK**.

To add weights to the GSLB services by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. Select the virtual server (for example vserver-GSLB-1) and click **Open**.
3. On the **Services** tab, in the **Weight** spin box, type or select the weight of a service (for example, next to service-GSLB-1, type 1).
4. Click **OK**.

To set dynamic weight by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Virtual Servers**.
2. Select the virtual server (for example vserver-GSLB-1) and click **Open**.
3. On the **Method and Persistence** tab, under **Method**, in **Dynamic Weight** drop-down list, select **SERVICEWEIGHT**.
4. Click **OK**.

Configuring for Disaster Recovery with Data Center Persistence

Data center persistence is required for web applications that require maintaining a connection with the same server instead of having the requests load balanced. For example, in an e-commerce portal, maintaining a connection between the client and the same server is critical. For such applications, HTTP redirect persistence can be configured in an active-active setup.

To configure GSLB for disaster recovery with data center persistence, you must first configure the basic GSLB set up and then configure HTTP redirect persistence.

First create two GSLB sites, local and remote. Next, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Next, create a load balancing virtual server with the same virtual server IP address as the GSLB service. Finally, duplicate the previous steps for the remote configuration, or configure the NetScaler appliance to autosynchronize your GSLB configuration.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure HTTP redirect precedence to enable data center persistence.

To configure HTTP redirect by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure HTTP redirect and verify the configuration:

- set gslb service <serviceName> **-sitePersistence** <sitePersistence> **-sitePrefix** <string>
- show gslb service <serviceName>

Example

set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1
show gslb service Service-GSLB-1

Parameters for configuring HTTP redirect

serviceName

The name of the GSLB service for which you are configuring site persistence.

sitePersistence

The type of site persistence being configured. Possible Values: ConnectionProxy, HTTPRedirect, None.

sitePrefix

When a service is bound to a GSLB virtual server, for each bound service-domain pair, a GSLB site domain is generated internally by concatenating the service's siteprefix and the domain name. If a special string, "NONE," is specified, the siteprefix string is not set.

To configure HTTP redirect by using the configuration utility

1. In the navigation pane, expand **GSLB** and click **Services**.
2. Select the GSLB service to be configured and click **Open**.
3. On the **Advanced** tab, under **Site Persistence options**, select the **HTTPRedirect** option.
4. In the **Site Prefix** text box, enter the site prefix (for example, vserver-GSLB-1).
5. Click **OK**.

Configuring GSLB for Proximity

When you configure GSLB for proximity, client requests are forwarded to the closest data center. The main benefit of the proximity-based GSLB method is faster response times resulting from the selection of the closest available data center. Such a deployment is critical for applications that require fast access to large volumes of data.

You can configure GSLB for proximity based on the round trip time (RTT), static proximity, or a combination of the two.

Configuring Dynamic Method (RTT)

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the NetScaler appliance probes the client's local DNS server and gathers RTT metric information. The NetScaler then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value

To configure GSLB for proximity with dynamic method, you must first configure the basic GSLB set up and then configure dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure the dynamic RTT method.

For details on how to configure the GSLB virtual server to use the dynamic RTT method for load balancing, see Configuring Dynamic RTT.

Configuring Static Proximity

The static proximity method for GSLB uses an IP address-based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The NetScaler appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the NetScaler appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

To configure GSLB for proximity with static proximity, you must first configure the basic GSLB set up and then configure static proximity.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure static proximity.

For details on how to configure the GSLB virtual server to use static proximity for load balancing, see Configuring Static Proximity.

Configuring Static Proximity and Dynamic RTT

You can configure the GSLB virtual server to use a combination of static proximity and dynamic RTT when you have some clients coming from an internal network like a branch office. You can configure GSLB such that the clients coming from the branch office or any other internal network are directed to a particular GSLB site that is geographically close to the client network. For all other requests, you can use dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure the GSLB virtual server to use static proximity for all traffic originating from an internal network and then use dynamic RTT for all other traffic.

For details on how to configure static proximity, see Configuring Static Proximity and for details on how to configure dynamic RTT, see Configuring Dynamic RTT.

EdgeSight Monitoring for NetScaler

Citrix EdgeSight® for NetScaler® is an application to monitor end-user experience with Web applications served in a NetScaler environment. The EdgeSight Monitoring application uses the HTML Injection feature of the NetScaler to provide data with which you can compare the performance of various Web applications across geographical locations.

For EdgeSight monitoring, register the NetScaler appliance with EdgeSight and enable applications in the NetScaler. The EdgeSight application processes the data and displays the information after aggregating it. You can view the data as charts, graphs, or tables.

When EdgeSight monitoring is enabled on a virtual server, NetScaler injects scripts into the responses sent to the clients. Execution or insertion of these scripts does not affect the response to the client. Data injected by the NetScaler is collected by the EdgeSight server through data collector services.

EdgeSight is an agentless application. The EdgeSight server from which you can monitor the user-experience is referred to as *EdgeSight UI server*.

Use the wizard to register the NetScaler appliance with the EdgeSight UI server, select the data collector services, and configure the rate at which data is injected into the response. For more information on the wizard, see Configuring EdgeSight Monitoring.

Note: The EdgeSight UI server has a data collector and a Web site. For better scalability, you can install multiple data collectors.

To use EdgeSight monitoring, enable EdgeSight monitoring on the load balancing or content switching virtual servers associated with the selected applications. For instructions, see Enabling an Application for EdgeSight Monitoring.

Configuring EdgeSight Monitoring for NetScaler

The configuration utility provides a wizard to assist you with configuration. The wizard for the configuration of EdgeSight monitoring for NetScaler guides you through the configuration procedure in simple steps. The wizard takes care of the following tasks:

- Enable the EdgeSight Monitoring (HTML Injection) feature.
- Specify the rate and frequency for injecting data.
- Bind the EdgeSight server/data collector services to the load balancing virtual server on the NetScaler.
- Enable the Web applications for EdgeSight monitoring.

Note: You can configure rate or frequency. If you specify the rate to be x, data is injected once after every x responses sent by the service. For example, if you specify the rate to be 500, NetScaler injects the data into the 1st reponse, 501st response, 1002nd response, and so on. If you specify frequency to be y, data is injected once in every y milliseconds.

To access the wizard from the NetScaler configuration utility and configure EdgeSight Monitoring

1. In the navigation pane, click **EdgeSight Monitoring**.
2. Click **EdgeSight for NetScaler Wizard**.
3. Follow the instructions presented by the wizard.

Configuring EdgeSight Monitoring for NetScaler

The configuration utility provides a wizard to assist you with configuration. The wizard for the configuration of EdgeSight monitoring for NetScaler guides you through the configuration procedure in simple steps. The wizard takes care of the following tasks:

- Enable the EdgeSight Monitoring (HTML Injection) feature.
- Specify the rate and frequency for injecting data.
- Bind the EdgeSight server/data collector services to the load balancing virtual server on the NetScaler.
- Enable the Web applications for EdgeSight monitoring.

Note: You can configure rate or frequency. If you specify the rate to be x, data is injected once after every x responses sent by the service. For example, if you specify the rate to be 500, NetScaler injects the data into the 1st reponse, 501st response, 1002nd response, and so on. If you specify frequency to be y, data is injected once in every y milliseconds.

To access the wizard from the NetScaler configuration utility and configure EdgeSight Monitoring

1. In the navigation pane, click **EdgeSight Monitoring**.
2. Click **EdgeSight for NetScaler Wizard**.
3. Follow the instructions presented by the wizard.

Enabling an Application for EdgeSight Monitoring

EdgeSight monitoring makes it possible to monitor the performance of an application from the end user's perspective.

Note: To enable an application to be monitored, EdgeSight monitoring must be enabled on the virtual servers.

EdgeSight monitoring is possible only if the response from the physical server is not compressed. Therefore, make sure that Compression is enabled globally on the NetScaler. When you use the wizard to configure EdgeSight monitoring, the wizard takes care of enabling compression and other necessary processing.

Note: Before enabling EdgeSight monitoring on a virtual server, make sure that you completed the configuration of EdgeSight monitoring through the wizard.

To enable EdgeSight monitoring on a load balancing or content switching virtual server by using the NetScaler configuration utility

Note: To enable EdgeSight monitoring on a content switching virtual server, follow the same procedure as given below.

1. In the navigation pane, expand **Load Balancing**.
2. Expand **Virtual Servers**.
3. Select the virtual server and click **Enable EdgeSight Monitoring**. You can select multiple servers to enable on all the selected servers.
4. Click **Yes** to accept the condition of not compressing the response from the server. The wizard takes care of the processing required to handle compression

Accessing the EdgeSight Monitoring Interface from NetScaler

You can view the data presented by the EdgeSight monitoring application from the NetScaler appliance. The configuration utility of the NetScaler displays a login screen to access the EdgeSight UI server and upon successful login, displays the data.

To access EdgeSight for NetScaler by using the NetScaler configuration utility

1. In the navigation pane, click **EdgeSight Monitoring**.
2. Click **Access EdgeSight for NetScaler**.
3. Enter the credentials for accessing the EdgeSight UI server. For more information on viewing the reports, see *EdgeSight Administration Guide* at http://support.citrix.com/article/CTX126418.

Variables Injected for EdgeSight Monitoring for NetScaler

For monitoring the end-user experience in the NetScaler environment, NetScaler injects scripts into the responses sent to the clients. A predefined set of variables are used in the scripts and the variables are evaluated at runtime. The following table describes the variables.

Table 1. Variables for Monitoring the NetScaler Performance

Name	Type	JavaScript type	Comment
SYS.IID	128-bit GUID structure	Windows format GUID	This is a GUID that uniquely identifies each NetScaler. The value of this variable remains constant across reboots. It is valid in both prebody and postbody.
HTTP.XID	128-bit GUID structure	Windows format GUID	This is a GUID that uniquely identifies each HTTP transaction (request/response). This variable is guaranteed to be unique even if the NetScaler is rebooted. It is valid in both the prebody and postbody.
SYS.UPTIME	32-bit integer	10-digit number	Gives the time in seconds, offset to UTC, that the NetScaler has been up. It is valid in both prebody and postbody.
HTTP.REQ.RECEIVE_TIME_BEG	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler received the first byte of a client request. It is valid in both prebody and postbody.

HTTP.REQ.RECEIVE_TIME_END	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler received the last byte of a client request. It is valid in both the prebody and postbody.
HTTP.REQ.SEND_TIME_BEG	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler forwarded the first byte of a request to the back-end server. It is valid in both prebody and postbody.
HTTP.REQ.SEND_TIME_END	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler forwarded the last byte of a request to the back-end server. It is valid in both prebody and postbody.
HTTP.RES.RECEIVE_TIME_BEG	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler received the first byte of a response from the back-end server. It is valid in both prebody and postbody.
HTTP.RES.RECEIVE_TIME_END	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler received the last byte of a response from the back-end server. It is valid only in postbody.
HTTP.RES.SEND_TIME_BEG	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler forwarded the first byte of response to the client. It is valid in both prebody and postbody.

HTTP.RES.SEND_TIME_END	64-bit integer	20-digit number	Gives the time, in microseconds, when NetScaler forwarded the last byte of a response to the client. It is valid only in postbody.
SYS.VSERVER	String (147 characters)	20-digit number	Gives the name, IP address, and port number of the virtual server that load balanced the request. It is valid in both prebody and postbody.
SYS.VSERVICE	String (147 characters)	20-digit number	Gives the name, IP address, and port number of the physical server that serviced the request. It is valid in both prebody and postbody.

Integrated Caching

The integrated cache provides in-memory storage on the Citrix® NetScaler® appliance and serves Web content to users without requiring a round trip to an origin server. For static content, the integrated cache requires little initial setup. After you enable the integrated cache feature and perform basic setup (for example, determining the amount of NetScaler appliance memory the cache is permitted to use), the integrated cache uses built-in policies to store and serve specific types of static content, including simple Web pages and image files. You can also configure the integrated cache to store and serve dynamic content that is usually marked as non-cacheable by Web and application servers (for example, database records and stock quotes).

When a request or response matches the rule (logical expression) specified in a built-in policy or a policy that you have created, the NetScaler appliance performs the action associated with the policy. By default, all policies store cached objects in and retrieve them from the Default content group, but you can create your own content groups for different types of content.

To enable the NetScaler appliance to find cached objects in a content group, you can configure selectors, which match cached objects against expressions, or you can specify parameters for finding objects in the content group. If you use selectors (which Citrix recommends), configure them first, so that you can specify selectors when you configure content groups. Next, set up any content groups that you want to add, so that they are available when you configure the policies. To complete the initial configuration, create policy banks by binding each policy to a global bind point or a virtual server, or to a label that can be called from other policy banks.

You can can tune the performance of the integrated cache, using methods such as pre-loading cached objects before they are scheduled to expire. To manage the handling of cached data once it leaves the NetScaler appliance, you can configure caching-related headers that are inserted into responses. The integrated cache can also act as a forward proxy for other cache servers.

Note: Integrated caching requires some familiarity with HTTP requests and responses. For information about the structure of HTTP data, see “Live HTTP Headers” at http://livehttpheaders.mozdev.org/.

How the Integrated Cache Works

The integrated cache monitors HTTP requests that flow through the Citrix® NetScaler® appliance and compares the requests with stored policies. Depending on the outcome, it either searches the cache for the response or forwards the request to the origin server. The integrated cache can also serve partial content from the cache in response to single byte-range requests.

Cached data can be compressed if the client accepts compressed content. You can configure expiration times for a content group, and you can selectively expire entries in a content group.

Data that is served from the integrated cache is a cache hit, and data served from the origin is a cache miss, as described in the following table.

Table 1. Cache Hits and Misses

Transaction Type	Specifies

Cache Hit	Responses that the NetScaler appliance serves from the cache, including: • Static objects, for example, image files and static Web pages • 200 OK pages • 203 Non-Authoritative Response pages • 300 Multiple Choices pages • 301 Moved Permanently pages • 302 Found pages • 304 Not Modified pages These responses are known as positive responses. The NetScaler appliance also caches the following negative responses: • 307 Temporary Redirect pages • 403 Forbidden pages • 404 Not Found pages • 410 Gone pages To further improve performance, you can configure the NetScaler appliance to cache additional types of content.
Storable Cache Miss	For a storable cache miss, the NetScaler appliance fetches the response from the origin server, and stores the response in the cache before serving it to the client.
Non-Storable Cache Miss	A non-storable cache miss is inappropriate for caching. By default, any response that contains the following status codes is a non-storable cache miss: • 201, 202, 204, 205, 206 status codes • All 4xx codes, except 403, 404 and 410 • 5xx status codes

Note: To integrate dynamic caching with your application infrastructure, use the XMLAPI to issue cache commands remotely. For example, you can configure triggers that expire cached responses when a database table is updated.

To ensure the synchronization of cached responses with the data on the origin server, you configure expiration methods. When the NetScaler appliance receives a request that matches an expired response, it refreshes the response from the origin server.

How the Integrated Cache Works

The integrated cache monitors HTTP requests that flow through the Citrix® NetScaler® appliance and compares the requests with stored policies. Depending on the outcome, it either searches the cache for the response or forwards the request to the origin server. The integrated cache can also serve partial content from the cache in response to single byte-range requests.

Cached data can be compressed if the client accepts compressed content. You can configure expiration times for a content group, and you can selectively expire entries in a content group.

Data that is served from the integrated cache is a cache hit, and data served from the origin is a cache miss, as described in the following table.

Table 1. Cache Hits and Misses

Transaction Type	Specifies

Cache Hit	Responses that the NetScaler appliance serves from the cache, including: • Static objects, for example, image files and static Web pages • 200 OK pages • 203 Non-Authoritative Response pages • 300 Multiple Choices pages • 301 Moved Permanently pages • 302 Found pages • 304 Not Modified pages These responses are known as positive responses. The NetScaler appliance also caches the following negative responses: • 307 Temporary Redirect pages • 403 Forbidden pages • 404 Not Found pages • 410 Gone pages To further improve performance, you can configure the NetScaler appliance to cache additional types of content.
Storable Cache Miss	For a storable cache miss, the NetScaler appliance fetches the response from the origin server, and stores the response in the cache before serving it to the client.
Non-Storable Cache Miss	A non-storable cache miss is inappropriate for caching. By default, any response that contains the following status codes is a non-storable cache miss: • 201, 202, 204, 205, 206 status codes • All 4xx codes, except 403, 404 and 410 • 5xx status codes

Note: To integrate dynamic caching with your application infrastructure, use the XMLAPI to issue cache commands remotely. For example, you can configure triggers that expire cached responses when a database table is updated.

To ensure the synchronization of cached responses with the data on the origin server, you configure expiration methods. When the NetScaler appliance receives a request that matches an expired response, it refreshes the response from the origin server.

Example of Dynamic Caching

Dynamic caching evaluates HTTP requests and responses based on parameter-value pairs, strings, string patterns, or other data. For example, suppose that a user searches for Bug 31231 in a bug reporting application. The browser sends the following request on the user's behalf:

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&Template=view&TableId=1000
Host: mycompany.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9) Gecko/2008052906 Firefox/3.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
. . .
```

In this example, GET requests for this bug reporting application always contain the following parameters:

- IssuePage
- RecordID
- Template
- TableId

GET requests do not update or alter the data, so you can configure these parameters in caching policies and selectors, as follows:

- You configure a caching policy that looks for the string mybugreportingsystem and the GET method in HTTP requests. This policy directs matching requests to a content group for bugs.
- In the content group for bugs, you configure a hit selector that matches various parameter-value pairs, including IssuePage, RecordID, and so on.

Note that a browser can send multiple GET requests based on one user action. The following is a series of three separate GET requests that a browser issues when a user searches for a bug based on a bug ID. **Bold** has been added for emphasis:

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&Template=view&TableId=1000
GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=viewbtns&RecordId=31231&TableId=1000
GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=viewbody&RecordId=31231&tableid=1000
```

To fulfill these requests, multiple responses are sent to the user's browser, and the Web page that the user sees is an assembly of the responses.

If a user updates a bug report, the corresponding responses in the cache should be refreshed with data from the origin server. The bug reporting application issues HTTP POST requests when a user updates a bug report. In this example, you configure the following to ensure that POST requests trigger invalidation in the cache:

- A request-time invalidation policy that looks for the string mybugreportingsystem and the POST HTTP request method, and directs matching requests to the content group for bug reports.
- An invalidation selector for the content group for bug reports that expires cached content based on the RecordID parameter. This parameter appears in all of the responses, so the invalidation selector can expire all relevant items in the cache.

The following excerpt shows a POST request that updates the sample bug report. **Bold** has been added for emphasis:

```
POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Opera 7.23 [en]\r\n
Host: mybugreportingsystem\r\n
Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23unowned%3Afalse%23submitter%3Afa
Cookie2: $Version=1\r\n
. . .
\r\n
ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=Update&CopyProjectId=0&ReloadForm=0
```

When the Citrix® NetScaler® appliance receives this request, it does the following:

- Matches the request with an invalidation policy.
- Finds the content group that is named in the policy.
- Applies the invalidation selector for this content group and expires all responses that match RecordID=31231.

When a user issues a new request for this bug report, the NetScaler appliance goes to the origin server for updated copies of all the responses that are associated with the report instance, stores the responses in the content group, and serves them to the user's browser, which reassembles the report and displays it.

Setting Up the Integrated Cache

To use the integrated cache, you must install the license and enable the feature. After you enable the integrated cache, the Citrix® NetScaler® appliance automatically caches static objects as specified by built-in policies and generates statistics on cache behavior. (Built-in policies have an underscore in the initial position of the policy name.)

Even if the built-in policies are adequate for your situation, you might want to modify the global attributes. For example, you might want to modify the amount of NetScaler appliance memory allocated to the integrated cache.

If you would like to observe cache operation before changing settings, see Displaying Cached Objects and Cache Statistics.

Note: The NetScaler cache is an in-memory store that is purged when you restart the appliance.

Installing the Integrated Cache License

An integrated cache license is required. For information about licenses, see information about obtaining NetScaler licenses athttp://support.citrix.com/article/ctx121062.

To install the license for the Integrated Caching feature

1. Obtain a license code from Citrix, go to the NetScaler command line, and log in.
2. At the NetScaler command line, copy the license file to the /nsconfig/license folder.
3. Reboot the NetScaler appliance by using the following command:

   reboot

Enabling or Disabling Integrated Cache

When you enable integrated caching, the NetScaler appliance begins caching server responses. If you have not configured any policies or content groups, the built in policies store cached objects in the Default content group.

To enable or disable integrated caching by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to enable or disable integrated caching:

- enable feature IC
- disable feature IC

To enable or disable integrated caching by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Configure basic features**.
3. In the **Configure Basic Features** dialog box, select the **Integrated Caching** check box to enable integrated caching, or clear the check box to disable it.
4. In the confirmation message box, click **OK**, and then click **Yes**.

Configuring Global Attributes for Caching

Global attributes apply to all cached data. You can specify the amount of NetScaler memory allocated to the integrated cache, Via header insertion, a criterion for verifying that a cached object should be served, the maximum length of a POST body permitted in the cache, whether to bypass policy evaluation for HTTP GET requests, and an action to take when a policy cannot be evaluated.

The cache memory capacity is limited only by the memory of the hardware appliance. Also, any packet engine (the central distribution hub of all incoming TCP requests) in the nCore NetScaler appliance is aware of objects cached by other packet engines in the nCore NetScaler appliance.

Note that the default global memory limit is 0. Therefore, even if Integrated Caching is enabled, the NetScaler appliance does not cache any objects. You must explicitly set the global memory limit when integrated caching is enabled.

In the nCore NetScaler appliance, when you reset the global memory limit to another positive number (for example, from 4000 MB to 6000 MB), the existing memory allocated to Integrated Caching will not be altered immediately. However, the output of the show cache parameter command shows that the memory usage limit is reset to 6000 MB, as follows:

```
> show cache parameter
    Integrated cache global configuration:
    Memory usage limit: 6000 MBytes
    Memory usage limit (active value): 4000 MBytes
    Maximum value for Memory usage limit: 7824 MBytes
    Via header: NS-CACHE-9.2: 230
    Verify cached object using: HOSTNAME
    Max POST body size to accumulate: 4096 bytes
    Current outstanding prefetches: 0
    Max outstanding prefetches: 4294967295
    Treat NOCACHE policies as BYPASS policies: YES
    Global Undef Action: NOCACHE
```

In the output, note that the memory usage limit’s active value (Memory usage limit (active value)) is an accurate indicator of the actual memory usage limit in effect.

To ensure that the change to the global memory limit is effected, save the new configuration and restart the appliance. The output of the show cache parameter command now indicates that the memory usage limit has indeed been updated to 6000 MB, as follows:

```
> show cache parameter
    Integrated cache global configuration:
    Memory usage limit: 6000 MBytes
```

Memory usage limit (active value): 6000 MBytes
Maximum value for Memory usage limit: 7824 MBytes
Via header: NS-CACHE-9.2: 230
Verify cached object using: HOSTNAME
Max POST body size to accumulate: 4096 bytes
Current outstanding prefetches: 0
Max outstanding prefetches: 4294967295
Treat NOCACHE policies as BYPASS policies: YES
Global Undef Action: NOCACHE

Note that the change is now effected and the active value of the memory usage limit is also updated to 6000 MB.

To configure global settings for caching by using the NetScaler command line

At the NetScaler command prompt, type:

set cache parameter [**-memLimit** <MBytes>] [**-via** <string>] [**-verifyUsing** <criterion>] [**-maxPostLen** <positiveInteger>] [**-prefetchMaxPending** <positiveInteger>] [**-enableBypass** (YES|NO)] [**-undefAction** (NOCACHE|RESET)]

Parameters for configuring global caching attributes

memlimit

Memory that the NetScaler appliance can dedicate to caching, up to 50% of available memory. A value of 0 prevents caching of any data, although the caching feature continues to run. Minimum value: 256 megabytes (MB). Maximum value: 7192 MB.

The minimum memory of 256 MB includes 240 MB of metadata, plus up to 100 MB for the URL, host, and additional HTTP data, plus the size of the cached object. The amount of storage is rounded up to the nearest 256 MB, to a maximum of approximately 1792 MB.

Memory usage limit cannot exceed memory available for the cache. The actual limit depends on the physical memory that is available, minus the memory required for other operations. In practice, the amount of memory that is available for caching can be less than half the total memory of the NetScaler appliance. The following are examples:

- NetScaler 12000 running 8.0: 1536 MB
- NetScaler 9010 running 8.0: 768 MB

To control memory use, you can also set a maximum and minimum response size for a content group. For more information, see About Content Groups.

Setting a very low memory usage limit can use up the memory quickly and prevent the cache from storing new objects.

via

String to insert in Via headers in the response. By default, a Via header is inserted in all cached responses. A Via header is not inserted for a cache miss.

Default value of the Via header is NS-CACHE-9.2:last octet of the NSIP. Maximum header length is 31.

For more information, see Inserting HTTP Headers at Response Time.

verifyUsing

Matches values in an HTTP request with a cached response based on one of the following possible values: HOSTNAME, HOSTNAME_AND_IP, DNS. Default: DNS.

If the values do not match, cached responses are not served. For example, if you specify HOSTNAME, the host name in the request URL must match the host name of the cached response. The DNS parameter performs DNS resolution on the host and IP address in the request URL.

This parameter offers an added security check and ensures that the request is from a valid source.

maxPostLen

Maximum number of POST body bytes to consider when evaluating parameters for a content group for which you have configured hit and invalidation parameters. Minimum value: 0. Maximum value: 131072.

Enter the length of the longest POST body that is to be evaluated at a time. For example, if you enter a value of 1,000, the NetScaler appliance evaluates the first 1,000 bytes of the POST body in the request, and then processes the next 1,000 bytes, and so on. This is also known as accumulation.

prefetchMaxPending

Maximum number of outstanding prefetches in the integrated cache.

enableBypass

Controls whether the NetScaler appliance evaluates request-time policies for HTTP GET requests. The NetScaler appliance always evaluates request-time policies for POST requests. Possible values: YES, NO. Default: YES.

- When Bypass is enabled (the default), the NetScaler appliance evaluates request-time policies before serving objects from the cache.
- Disable Bypass only if you want to avoid evaluation of request-time policies for anything other than POST requests. NetScaler appliance searches the cache and serves cached responses even if there is a matching NOCACHE policy. If the response is not in the cache, NetScaler appliance fetches it from the origin server, and performs the usual response-time policy processing.

Note: Bypass is always enabled when you create a selector-based or a parameterized group, even if the global setting is NO. Enabling bypass is always a safe option. The only reason to disable bypass is for efficiency, but this option is not available for parameterized or selector-based caching.

undefAction

Action to take when a policy cannot be evaluated. Possible values: NOCACHE, RESET. Default: NOCACHE.

To configure global settings for caching by using the configuration utility

1. In the navigation pane, click **Integrated Caching**.
2. In the details pane, click **Change cache settings**.
3. In the **Cache Global Settings** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring global attributes for caching” as shown:
   - **Memory Usage Limit (MB)**—memlimit
   - **Via Header**—via
   - **Maximum Post body length to be Cached**—maxPostLen
   - **Global Undefined-Result Action**—undefAction
   - **Bypass**—enableBypass
   - **Verify cached objects using**—verifyUsing
   - **Prefetches**—prefetchMaxPending
4. Click **OK**.

Built-in Content Group, Pattern Set, and Policies for the Integrated Cache

The Citrix® NetScaler® appliance includes a built-in integrated caching configuration that you can use for caching content. The configuration consists of a content group called `ctx_cg_poc`, a pattern set called `ctx_file_extensions`, and a set of integrated cache policies. In the content group `ctx_cg_poc`, only objects that are 500 KB or smaller are cached. The content is cached for 86000 seconds, and the memory limit for the content group is 512 MB. The pattern set is an indexed array of common file extensions for file-type matching.

The following table lists the built-in integrated caching policies. By default, the policies are not bound to any bind point. You must bind them to a bind point if you want the NetScaler appliance to evaluate traffic against the policies. The policies cache objects in the `ctx_cg_poc` content group.

Table 1. Built-in Integrated Caching Policies

me	Policy rule
	`HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_INDEX(\"ctx_file_extensions\").BETWEEN(101,150)`
	`HTTP.REQ.URL.ENDSWITH(\".css\")`
	`HTTP.REQ.URL.ENDSWITH(\".pdf\")`
	`HTTP.REQ.URL.ENDSWITH(\".js\")`
s	`HTTP.RES.HEADER(\"Content-Type\").CONTAINS(\"application/x-javascript\")`
	`TRUE`

Configuring Selectors and Basic Content Groups

You can configure selectors and apply them to content groups. When you add a selector to one or more content groups, you specify whether the selector is to be used for identifying cache hits or identifying cached objects to be invalidated (expired). Selectors are optional. Alternatively, you can configure content groups to use hit parameters and invalidation parameters. However, Citrix recommends that you configure selectors.

After configuring selectors, or deciding to use parameters instead, you are ready to set up a basic content group. After creating the basic content group, you need to decide how objects should be expired from the cache, and configure cache expiration. You can further modify the cache as described in Improving Cache Performance and Configuring Cookies, Headers, and Polling, but you might first want to configure caching policies.

Note: Content group parameters and selectors are used only at request time, and you typically associate them with policies that use MAY_CACHE or MAY_NOCACHE actions.

Advantages of Selectors

A selector is a filter that locates particular objects in a content group. If you do not configure a selector, the Citrix® NetScaler® appliance looks for an exact match in the content group. This can lead to multiple copies of the same object residing in a content group. For example, a content group that does not have a selector may need to store URLs for host1.domain.com\mypage.htm, host2.domain.com\mypage.htm, and host3.domain.com\mypage.htm. In contrast, a selector can match just the URL (mypage.html, using the expression http.req.url) and the domain (.com, using the expression http.req.hostname.domain), allowing the requests to be satisfied by the same URL.

Selector expressions can perform simple matching of parameters (for example, to find objects that match a few query string parameters and their values). A selector expression can use Boolean logic, arithmetic operations, and combinations of attributes to identify objects (for example, segments of a URL stem, a query string, a string in a POST request body, a string in an HTTP header, a cookie). Selectors can also perform programmatic functions to analyze information in a request. For example, a selector can extract text in a POST body, convert the text into a list, and extract a specific item from the list.

For more information about expressions and what you can specify in an expression, see the discussion of advanced expressions in the *Citrix NetScaler Policy Configuration and Reference Guide* at .

Using Parameters Instead of Selectors

Although Citrix recommends the use of selectors with a content group, you can instead configure hit parameters and invalidation parameters. For example, suppose that you configure three hit parameters in a content group for bug reports: BugID, Issuer, and Assignee. If a request contains BugID=456, with Issuer=RohitV and Assignee=RobertS, the NetScaler appliance can serve responses that match these parameter-value pairs.

Invalidation parameters in a content group expire cached entries. For example, suppose that BugID is ab invalidation parameter and a user issues a POST request to update a bug report. An invalidation policy directs the request to this content group, and the invalidation parameter for the content group expires all cached responses that match the BugID value. (The next time a user issues a GET request for this report, a caching policy can enable the NetScaler appliance to refresh the cached entry for the report from the origin server.)

Note that the same parameter can be used as a hit parameter or an invalidation parameter.

Content groups extract request parameters in the following order:

- URL query
- POST body
- Cookie header

After the first occurrence of a parameter, regardless of where it occurred in the request, all its subsequent occurrences are ignored. For example, if a parameter exists both in the URL query and in the POST body, only the one in the URL query is considered.

If you decide to use hit and invalidation parameters for a content group, configure the parameters when you configure the content group.

Note: Citrix recommends that you use selectors rather than parameterized content groups, because selectors are more flexible and can be adapted to more types of data.

Configuring a Selector

A content group can use a hit selector to retrieve cache hits or use an invalidation selector to expired cached objects and fetch new ones from the origin server.

A selector contains a name and a logical expression, called an *advanced expression*.

For more information about advanced expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

To configure a selector, you assign it a name and enter one or more expressions. As a best practice, a selector expression should include the URL stem and host, unless there is a strong reason to omit them.

To configure a selector by using the NetScaler command line

At the NetScaler command prompt, type:

add cache selector <selectorName> ("<expression>" ...)

For information about configuring the expression or expressions, see To configure a selector expression by using the NetScaler command line.

Examples

```
add cache selector product_selector "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"Bat
add cache selector batch_selector "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"Batch
add cache selector product_id_selector "http.req.url.query.value(\"ProductId\")"
add cache selector batchnum_selector "http.req.url.query.value(\"BatchNum\")" "http.req.url.query.value(\"
add cache selector batchid_selector "http.req.url.query.value(\"depotLocation\")" "http.req.url.query.value(
```

To configure a selector by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Cache Selectors**.
2. In the details pane, click **Add**.
3. In the **Create Cache Selector** dialog box, in the **Name** text box, type the name of the selector (for example, **myDatabase_hitSelector**).
4. In the **Expressions** box, enter the expression that you want to use for matching items in the cache (as described in “To configure a policy or selector expression by using the configuration utility”).
5. Click **Create**, and then click **Close**.

About Content Groups

A content group is a container for cached objects that can be served in a response. When you first enable the integrated cache, cacheable objects are stored in a content group named Default. You can create new content groups that have unique properties. For example, you can define separate content groups for image data, bug reports, and stock quotes, and you can configure the stock quote content group to be refreshed more often than the other groups.

You can configure expiration of an entire content group or selected entries in a content group.

The data in a content group can be static or dynamic, as follows:

- **Static content groups**. Finds an exact match between the URL stem and host name on the request and the URL stem and host name of the response.
- **Dynamic content groups**. Looks for objects that contains particular parameter-value pairs, arbitrary strings, or string patterns. Dynamic content groups are useful when caching data that is updated frequently (for example, a bug report or a stock quote).

Process overview: Serving a hit from a content group

1. A user enters search criteria for an item, such as a bug report, and clicks the Find button in an HTML form.
2. The browser issues one or more HTTP GET requests. These requests contain parameters (for example, the bug owner, bug ID, and so on).
3. When the NetScaler appliance receives the requests, it searches for a matching policy, and if it finds a caching policy that matches these requests, it directs the requests to a content group.
4. The content group looks for appropriate objects in the content group, usually based on criteria that you configure in a selector.

   For example, the content group can retrieve responses that match NameField=username and BugID=ID.
5. If it finds matching objects, the NetScaler appliance can serve them to the user's browser, where they are assembled into a complete response (for example, a bug report).

Example: Invalidating an object in a content group

1. A user modifies data (for example, the user modifies the bug report and clicks the Submit button).

2. The browser sends this data in the form of one or more HTTP requests. For example, it can send a bug report in the form of several HTTP POST requests that contain information about the bug owner and bug ID.

3. The NetScaler appliance matches the requests against invalidation policies. Typically, these policies are configured to detect the HTTP POST method.

4. If the request matches an invalidation policy, the NetScaler appliance searches the content group that is associated with this policy, and expires responses that match the configured criteria for invalidation.

   For example, an invalidation selector can find responses that match NameField=username and BugID=ID.

5. The next time the NetScaler appliance receives a GET request for these responses, it fetches refreshed versions from the origin server, caches the refreshed responses, and serves these responses to the user's browser, where they are assembled into a complete bug report.

Setting Up a Basic Content Group

By default, all cached data is stored in the Default content group. You can configure additional content groups and specify these content groups in one or more policies.

You can configure content groups for static content, and you must configure content groups for dynamic content. You can modify the configuration of any content group, including the Default group.

To set up a basic content group by using the NetScaler command line

At the NetScaler command prompt, type:

add cache contentgroup <name> (**-hitSelector** <hitSelectorName> **-invalSelector** <invalidationSelectorName> | **-hitParams** <hitParamName> **-invalParams** <invalidationParamName>)[**-relExpiry** <sec> | **-relExpiryMilliSec** <msec>] [**-heurExpiryParam** <positiveInteger>]

Examples

add cache contentgroup Products_Details –hitSelector product_selector –invalSelector id_selector
add cache contentgroup bugrep -hitParams IssuePage RecordID Template TableId -invalParams RecordID -rel

Parameters for configuring a basic content group

name

The name of this content group, up to 31 characters.

hitSelector and invalSelector

Search filters for the content group, as described in Configuring a Selector.

hitParams and invalParams

As an alternative to selectors, you can configure hit and invalidation parameters in the content group definition.

Note: Citrix recommends that you configure selectors as a best practice. However, if instead you configure hit and invalidation parameters, you can configure up to 128 hit parameters with a maximum length of 4095, and up to a maximum of 8 invalidation parameters with a maximum length of 255.

For caching parameters, determine whether searches are case sensitive. For invalidation parameters, determine whether the target host should be considered. For both caching and invalidation parameters, determine whether the caching module should also evaluate cookies in the request.

relExpiry and -relExpiryMilliSec

The relative expiry time in seconds. Default value: VAL_NOT_SET. Minimum value: 0. Maximum value: 31536000.

heurExpiryParam

The heuristic expiry time, in percent of the duration since the object was last modified. Default value: VAL_NOT_SET. Minimum value: 0. Maximum value: 100.

To set up a basic content group by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click **Add**, or select the name of an existing content group, and then click **Open**.
3. In the **Create Cache Content Group** or the **Configure Cache Content Group** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a basic content group” as shown:
   - **Name**—name
   - **Hit selector**—hitSelector
   - **Invalidation selector**—invalSelector
   - **Hit parameters**—hitParams
   - **Invalidation parameters**—invalParams
   - **Expiry content after**—relExpiry and relExpiryMilliSec
   - **Heuristic**—heurExpiryParam
4. Click **Create** or **OK**.

Expiring or Flushing Cached Objects

If a response does not have an Expires header or a Cache-Control header with an expiration time (Max-Age or Smax-Age), you must expire objects in a content group by using one of the following methods:

- Configure content group expiration settings to determine whether and how long to keep the object.
- Configure an invalidation policy and action for the content group. For more information, see Configuring Policies for Caching and Invalidation.
- Expire the content group or objects within it manually.

After a cached response expires, the NetScaler appliance refreshes it the next time the client issues a request for the response. By default, when the cache is full, the NetScaler appliance replaces the least recently used response first.

The following list describes methods for expiring cached responses using settings for a content group. Typically, these methods are specified as a percent or in seconds:

- **Manual**. Manually invalidate all responses in a content group or all responses in the cache.
- **Response-based**. Specific expiration intervals for positive and negative responses. Response-based expiry is considered only if the Last-Modified header is missing in the response.
- **Heuristic expiry**. For responses that have a Last-Modified header, heuristic expiry is a percentage of the time since the response was modified (calculated as current time minus the Last-Modified time, multiplied by the heuristic expiry value). For example, if a Last-Modified header indicates that a response was updated 2 hours ago, and the heuristic expiry setting is 10%, cached objects expire after 0.2 hours. This method assumes that frequently updated responses need to be expired more often.
- **Absolute or relative**. Specify an exact (absolute) time when the response expires every day, in HH:MM format, local time or GMT. Local time may not work in all time zones.

  Relative expiration specifies a number of seconds or milliseconds from the time a cache miss causes a trip to the origin server to the expiration of the response. If you specify relative expiration in milliseconds, enter a multiple of 10. This form of expiration works for all positive responses. Last-Modified, Expires, and Cache-Control headers in the response are ignored.

  Absolute and relative expiration override any expiration information in the response itself.
- **On download**. The option Expire After Complete Response Received expires a response as soon as it is downloaded. This is useful for frequently updated responses, for example, stock quotes. By default, this option is disabled.

Enabling both Flash Cache and Expire After Complete Response Received accelerates the performance of dynamic applications. When you enable both options, the NetScaler appliance fetches only one response for a block of simultaneous requests.

For more information, see Queuing Requests to the Cache.

- **Pinned**. By default, when the cache is full the NetScaler appliance replaces the least recently used response first. The NetScaler appliance does not apply this behavior to content groups that are marked as pinned.

If you do not configure expiration settings for a content group, the following are additional options for expiring objects in the group:

- Configure a policy with an INVAL action that applies to the content group.
- Enter the names of content groups when configuring a policy that uses an INVAL action.

How Expiration Methods Are Applied

Expiration works differently for positive and negative responses. Positive and negative responses are described in the table, Cache Hits and Misses.

The following are rules of thumb for understanding the expiration method that is applied to a content group:

- You can control whether the NetScaler appliance evaluates response headers when deciding whether to expire an object.
- Absolute and relative expiration cause the NetScaler appliance to ignore the response headers (they override any expiration information in the response).
- Heuristic expiration settings and “Weak Positive” and “Weak Negative” expiration (labeled as **Default** values in the configuration utility) cause the NetScaler appliance to examine the response headers. These settings work together as follows:
  - The value in an Expires or Cache-Control header overrides these content group settings.
  - For positive responses that lack an Expires or Cache-Control header but have a Last-Modified header, the NetScaler appliance compares heuristic expiration settings with the header value.
  - For positive responses that lack an Expires, Cache-Control, or Last-Modified header, NetScaler appliance uses the “weak positive” value.
  - For negative responses that lack an Expires or Cache-Control header, NetScaler appliance uses the “weak negative” value.

A complete list of expiration methods is provided in the table Parameters for configuring content group expiration. The following table describes how these methods are applied.

Table 1. Expiration of Positive and Negative Responses

Response Type	Expiration Header Type	Content Group Setting	Period the Object Remains in the Cache
Positive	any header	Expire Content After (relExpiry) with no other settings	Use the value of the Expire Content After setting.
Positive	any header	Expire Content At (absExpiry) with no other settings	Subtract current date from the value of the Expire Content At setting.
Positive	any header	Expire Content After (relExpiry) and Expire content at (absExpiry)	Use the smaller of the two values for the content group settings. See the previous rows in this table.
Positive	Last-Modified (with any other headers)	Heuristic (heurExpiry Param) with any other setting	Subtract the Last-Modified date from the current date, multiply the result by the value of the heuristic expiry setting, and then divide by 100.
Positive	Last-Modified (with any other headers)	Default (positive) (weakPosRel Expiry) and no other setting	Use the value of the Default (positive) expiry setting.
Positive	Expires or Cache-Control: Max-Age header is present Last-Modified header is absent	Heuristic (heurExpiry Param), Default (positive) (weakPosRel Expiry), or both	Subtract the current date from the Expires or the Cache-Control:Max-Age date.
Positive	no caching headers	Default (positive) (weakPosRel Expiry) and any other expiration setting.	Use the value of the Default (positive) setting.

Positive	no caching headers	Heuristic (heurExpiry Param) is present Default (positive) (weakPosRel Expiry) setting is absent	If the Last-Modified header is absent, the response is not cached or it is cached with an Already Expired status. If the Last-Modified header is present, use the heuristic expiry value.
Negative	Expires or Cache-Control:Max-Age	Expire Content After (relExpiry), Expire Content At (absExpiry), or both settings	Subtract the current date from the value of the Expires header, or use the value of the Cache-Control:Max-Age header.
Negative	Expires or Cache-Control headers are absent	Expire Content After (relExpiry), Expire Content At (absExpiry), or both settings	Response is not cached, or is cached with an Already Expired status.
Negative	Expires or Cache-Control:Max-Age	Any setting	Subtract the current date from the Expires or Cache-Control:Max-Age date.
Negative	Expires and Cache-Control:Max-Age headers are absent	Default (negative) (weakNegRel Expiry)	Use the value of the Default (negative) setting.
Negative	Expires and Cache-Control:Max-Age headers are absent	Any setting other than Default (negative) (weakNegRel Expiry)	Object is not cached or is cached with an Already Expired status.

Expiring a Content Group Manually

You can manually expire all of the entries in a content group.

To manually expire all responses in a content group by using the NetScaler command line

At the NetScaler command prompt, type:

expire cache contentGroup <contentGroupName>

To manually expire all responses in a content group by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group that you want to invalidate, and click **Invalidate**.
3. In the **Invalidate Selected Cache Content Group** dialog box, click **Expire**, and then click **OK**.

To manually expire all responses in the cache by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click **Invalidate All**.
3. In the **Invalidate All Content Groups** dialog box, click **Expire All**, and then click **OK**.

Configuring Periodic Expiration of a Content Group

You can configure a content group so that it performs selective or full expiration of its entries. The expiration interval can be fixed or relative.

To configure content group expiration by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <name>
(**-relExpiry**|**-relExpiryMilliSec**|**-absExpiry**|**-absExpiryGMT**|
-heurExpiryParam|**-weakPosRelExpiry**|**-weakNegRelExpiry**| **expireAtLastBye**)
<expirationValue>

Parameters for configuring content group expiration

name

The name of the content group name whose attributes will be changed.

-relExpiry

-relExpiry MilliSec

Relative expiration time for cached responses in seconds and milliseconds.

The minimum is 0 and the maximum is 3153600 seconds.

-absExpiry

-absExpiry GMT

Time of day when the cached responses expire. You can specify up to four times per day in local time or GMT.

expireAtLastByte

Enables expiration immediately after the response is downloaded. Possible values: YES, NO. Default value: NO.

-heurExpiry Param

Heuristic expiration time of a cached response, calculated as a percent of the time since the response was last modified. Default: 0. Minimum: 0. Maximum:100.

-weakPosRel Expiry

Expiration time for positive responses in a content group. For more information, see Cache Hits and Misses.

Default: 3600 seconds. Minimum: 0. Maximum: 31536000 seconds.

Note that other properties of the response take precedence over this setting. For example, the relExpiry setting for the content group has a higher priority. Higher priority is also assigned if the response does not contain Expires or Last-Modified headers.

-weakNegRel Expiry

Expiration time for negative responses in a content group, as described in the table Cache Hits and Misses.

Minimum:0. Maximum: 31536000 seconds.

If the origin server has already set a Cache-Control header with an expiration setting for the negative response, the header determines the expiration. Higher priority is also given to Expires or Last-Modified headers. If there is no header for expiration, the weak negative relative expiration value is used.

-expireAt LastByte

Enables expiration immediately after the response is downloaded. Applicable only to positive responses. For details, Queuing Requests to the Cache.

To configure content group expiration by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to set the heuristic expiry parameter, and then click **Open**.
3. In the **Configure Cache Content Group** dialog box, on the **Expiry Method** tab, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an expiration method" as shown:
   - **Expire content after**—relExpiry or relExpiryMilliSec
   - **Expire content at (HH:MM)**—absExpiry or absExpiryGMT
   - **Expire after complete response received**—expireAtLastByte
   - **With Last-modified header** - Heuristic parameter—heurExpiryParam
   - **For positive (non-error) responses e.g. 2xx 3xx**—weakPosRelExpiry
   - **For negative (error) responses e.g. 4xx 5xx**—weakNegRelExpiry
4. Click **Create**.

Expiring Individual Responses

Expiring a response forces the NetScaler appliance to fetch a refreshed copy from the origin server. Responses that do not have validators, for example, ETag or Last-Modified headers, cannot be revalidated. As a result, flushing these responses has the same effect as expiring them.

To expire a cached response in a content group for static data, you can specify a URL that must match the stored URL.["byte by byte" is redundant -megha] If the cached response is part of a parameterized content group, you must specify the group name as well as the exact URL stem. The host name and the port number must be the same as in the host HTTP request header of the cached response. If the port is not specified, port 80 is assumed.

To expire individual responses in a content group by using the NetScaler command line

At the NetScaler command prompt, type:

expire cache object **-url** <URL> **-host** <hostName> [**-port** <port>] [**-groupName** <contentGroupName>] [**-httpMethod** GET|POST]

To expire individual responses in a content group by using the NetScaler command line (selector-based)

At the NetScaler command line, type the following command:

expire cache object **-locator** <positiveInteger>

Parameters for expiring individual responses

url

URL stem of the response to be expired, for example, /mycompany/mydocuments.

host

Host name of origin server from which the response was fetched.

port

Listen port of origin server from which the response was fetched. Default value: 80.

groupName

Name of content group where a parameterized cached response is stored.

httpMethod

Possible values: GET and POST. Default: GET.

locator

Unique identification number of the cached response, expressed as a positive hexadecimal number. Maximum: 1.84467440737096e+19.

To expire a cached response by using the configuration utility

1. View the cached response. For more information, see Displaying Cached Objects and Cache Statistics.
2. Click the response you want to expire.
3. Click **Expire**.

To expire a response by using the Lookup tool (selector-based)

1. Find the response that you want to expire. For more information, see Finding Particular Cached Responses.
2. Click **Expire**.

Flushing Responses in a Content Group

You can remove, or flush, all responses in a content group, some responses in a group, or all responses in the cache. Flushing a cached response frees up memory for new cached responses.

Note: To flush responses for more than one object at a time, use the configuration utility method. The NetScaler command line does not offer this option.

To flush responses from a content group by using the NetScaler command line

At the NetScaler command prompt, type:

flush cache contentGroup <name> [**-query** <queryString> | [**-selectorValue** <selectorExpressionIDList> **-host** <hostName>]]

Parameters for flushing a content group

name

The read-only content group name.

query

If the content group has an invalidation parameter, and the option Invalidate Objects Belonging to Target Host is not selected, you can flush objects using a query string. For example, suppose that you configure a name invalidation parameter and the content group contains the following objects:

```
0x00000007e12800000005  mygroup GET    //192.168.100.116:80/index.html?name=alena
0x0000000d099500000006  mygroup GET    //192.168.100.116:80/index.html?name=john
```

To flush the first object, you would enter name=alena in this field.

selectorValue

If content group has an invalidation selector, you have the option to expire objects using the selector value. For example, suppose that you have an invalidation selector with a value of http.req.url and the following stored objects:

```
0x0000000a436c00000004  mygroup GET     //192.168.100.116:80?_1=/index.html
0x0000000c0f1a00000003  mygroup GET     //192.168.100.116:80?_1=/cgi-bin/rfc/nph-200.pl
```

To flush both objects, you would enter a value of _1. To flush only the first object, you would enter _1=/index.html.in this field.

host

If the content group has an invalidation parameter, and the option Invalidate Objects Belonging to Target Host is selected, you can flush objects using a query string and host. For example, suppose that the following content group has two objects that are identical with the exception of the host value:

```
0x00000003ca4c00000007  mygroup GET     //mycompany.com:80/index.html?name=john
0x00000005ebbf00000008  mygroup GET     //mycompany2.com:80/index.html?name=john
```

To expire only the first object, you would enter name=john in the Query field and mycompany in the Host field.

To flush responses from a content group by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In details pane, flush the responses as follows:
   - To flush all responses in all content groups, click **Invalidate All**. In the **Invalidate All Content Groups**, click **Flush All**.
   - To flush responses in a particular content group, select the content group that you want to invalidate, and then click **Invalidate**. In the **Invalidate Selected Cache Content Group** dialog box, click **Flush**.
3. Click **OK**, and then click **Close**.

Note: If this content group uses a selector, you can selectively flush responses by entering a string in the **Selector value** text box, entering a host name in the **Host** text box. Then click **Flush** and **OK**. The **Selector value** can be a query string of up to 2319 characters that is used for parameterized invalidation.

If the content group uses an invalidation parameter, you can selectively flush responses by entering a string in the **Query** field.

If the content group uses an invalidation parameter and Invalidate objects belonging to target host is configured, enter strings in the **Query** and **Host** fields.

To flush a cached response by using the NetScaler command line

At the NetScaler command prompt, type:

flush cache object **-locator** <positiveInteger> | **-url** <URL> **-host** <hostName> [**-port** <port>] [**-groupName** <contentGroupName>] [**-httpMethod** GET|POST]

To flush a cached response by using the configuration utility

1. Find the cached response. For more information, see Configuring Global Attributes for Caching.
2. Select the response that you want to expire.
3. Click **Flush**.

Deleting a Content Group

You can remove a content group if it is not used by any policy that stores responses in the cache. If the content group is bound to a policy, you must first remove the policy. Removing the content group removes all responses stored in that group.

You cannot remove the Default, BASEFILE, or Deltajs group. The Default group stores cached responses that do not belong in any other content group.

To delete a content group by using the NetScaler command line

At the NetScaler command prompt, type:

rm cache contentgroup <contentGroupName>

To delete a content group by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and click **Content Groups**.
2. In the details pane, click the content group name that you want to remove, and click **Remove**.
3. In the **Remove** message box, click **Yes**.

Configuring Policies for Caching and Invalidation

Policies enable the integrated cache to determine whether to try to serve a response from the cache or the origin. The Citrix® NetScaler® appliance provides built-in policies for integrated caching, and you can configure additional policies. When you configure a policy, you associate it with an action. An action either caches the objects to which the policy applies or invalidates (expires) the objects. Typically, you based caching policies on information in GET and POST requests. You typically base invalidation policies on the presence of the POST method in requests, along with other information. You can use any information in a GET or POST request in a caching or an invalidation policy.

You can view some of the built-in policies in the integrated cache's Policies node in the configuration utility. The built-in policy names begin with an underscore (_).

Actions determine what the NetScaler appliance does when traffic matches a policy. The following actions are available:

- **Caching actions**. Policies that you associate with the CACHE action store responses in the cache and serve them from the cache.
- **Invalidation actions**. Policies that you associate with the INVAL action immediately expire cached responses and refresh them from the origin server. Note that for Web-based applications, invalidation policies often evaluate POST requests.
- **"Do not cache" actions**. Policies that you associate with a NOCACHE action never store objects in the cache.
- **"Provisionally cache" actions**. Policies that you associate with a MAYCACHE or MAYNOCACHE action depend on the outcome of additional policy evaluations.

Although the integrated cache does not store objects specified by the LOCK method, you can invalidate cached objects upon receipt of a LOCK request. For invalidation policies only, you can specify LOCK as a method by using the expression `http.req.method.eq("lock")`. Unlike policies for GET and POST requests, you must enclose the LOCK method in quotes because the NetScaler appliance recognizes this method name as a string only.

After you create a policy, you bind it to a particular point in the overall processing of requests and responses. Although you create a policy before binding it, you should understand how the bind points affect the order of processing before you create your policies.

The policies bound to a particular bind point constitute a policy bank. You can use goto expressions to modify the order of execution in a policy bank. You can also invoke policies in other policy banks. In addition, you can create labels and bind policies to them. Such a label is not associated with a processing point, but the policies bound to it can be invoked from other policy banks.

Actions to Associate with Integrated Caching Policies

The following table describes actions for integrated caching policies.

Table 1. Actions That You Can Associate with an Integrated Caching Policy

Action	Specifies
CACHE	Serves a response from the cache if the response has not expired. If the response must be fetched from the origin server, the NetScaler appliance caches the response before serving it. Even data that is updated and accessed frequently can be cached. For example, stock quotes are updated frequently, but they can be cached so that they can be served quickly to multiple users. If necessary, cached data can be refreshed immediately after it is downloaded. A CACHE action can be overridden by built-in policies.
NOCACHE	Always fetches the response from the origin server and marks the response as non-storable. You typically configure NOCACHE policies for data that is sensitive or personalized.
MAY_CACHE	Used in a request-time policy, this setting provisionally enables a response to be stored in a content group, pending evaluation of response-time policies. The following are possible: • If a matching response-time policy has a CACHE action but does not specify a content group, the response is stored in the Default group unless built-in policies override this policy. • If a matching response-time policy has a CACHE action and specifies the same content group as the one in the request-time policy, the response is stored in the named content group unless built-in policies override this policy. • If a matching response-time policy has a CACHE action but specifies a different content group from the one in the request-time policy, a NOCACHE action is applied. • If a matching response-time policy has a NOCACHE action, perform a NOCACHE action. • If there is no matching response-time policy, a CACHE action is applied, unless a built-in policy overrides this policy.

MAY_NOCACHE	For a request-time policy, this setting provisionally prevents caching the response. At response time, one of following actions is taken: • If no response-time policy matches the request, the final action is NOCACHE. • If a matching response-time policy contains a CACHE action, the final action is CACHE, unless built-in policies override this policy. • If a matching response-time policy contains a NOCACHE action, the final action is NOCACHE. • If a matching response-time policy has a CACHE action but does not specify a content group, the final action is to CACHE the response in the Default content group, unless built-in policies override this policy.
INVAL	Expires cached responses. Depending on how the policy and the content group are configured, all responses in one or more content groups are expired, or selected objects in the content group are expired. **Note:** You can specify INVAL actions in request-time policies only.

Bind Points for a Policy

You can bind the policy to one of the following bind points:

- **A global policy bank**. These are the request-time default, request-time override, response-time default, and response-time override policy banks, as described in Order of Policy Evaluation.
- **A virtual server**. Policies that you bind to a virtual server are processed after the global override policies and before the global default policies, as described in Order of Policy Evaluation. Note that when binding a policy to a virtual server, you bind it to either request-time or response-time processing.
- **An ad-hoc policy label**. A policy label is a name assigned to a policy bank. In addition to the global labels, the integrated cache has two built-in custom policy labels:
  - **_reqBuiltinDefaults**. This policy label, by default, is invoked from the request-time default policy bank.
  - **_resBuiltinDefaults**. This policy label, by default, is invoked from the response-time default policy bank.

  You can also define new policy labels. Policies bound to a user-defined policy label must be invoked from within a policy bank for one of the built-in bind points. For more information about creating a policy label, see Configuring a Policy Label in the Integrated Cache. For more information about policy label invocation, see Configuring a Policy Bank for Caching.

Important: You should bind a policy with an INVAL action to a request-time override or a response-time override bind point. To delete a policy, you must first unbind it.

Order of Policy Evaluation

For an advanced policy to take effect, you must ensure that the policy is invoked at some point during the NetScaler appliance's processing of traffic. To specify the invocation time, you associate the policy with a bind point. The following are the bind points, listed in order of evaluation:

- **Request-time override**. If a request matches a request-time override policy, by default request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time load balancing virtual server**. If policy evaluation cannot be completed after all the request-time override policies are evaluated, the NetScaler appliance processes request-time policies that are bound to load balancing virtual servers. If the request matches one of these policies, evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time content switching virtual server**. Policies that are bound to this bind point are evaluated after request-time policies that are bound to load balancing virtual servers.

- **Request-time default**. If policy evaluation cannot be completed after all request-time, virtual server-specific policies are evaluated, the NetScaler appliance processes request-time default policies. If the request matches a request-time default policy, by default request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.

- **Response-time override**. Similar to request-time override policy evaluation.

- **Response-time load balancing virtual server**. Similar to request-time virtual server policy evaluation.

- **Response-time content switching virtual server**. Similar to request-time virtual server policy evaluation.

- **Response-time default**. Similar to request-time default policy evaluation.

You can associate multiple policies with each bind point. To control the order of evaluation of the policies associated with the bind point you configure a priority level. In the absence of any other flow control information, policies are evaluated according to priority level, starting with the lowest numeric priority value.

After all integrated caching policies have been evaluated, if there are conflicting actions specified in request-time and response-time policies, the NetScaler appliance determines the final action as specified in the table, Actions That You Can Associate with an Integrated Caching Policy.

Note: Request-time policies for POST data or cookie headers must be invoked during request-time override evaluation, because the built-in request-time policies in the integrated cache return a NOCACHE action for POST requests and a MAY_NOCACHE action for requests with cookies. Note that you would associate MAY_CACHE or MAY_NOCACHE actions with a request-time policy that points to a parameterized content group. The response time policy determines whether the transaction is stored in the cache.

Configuring a Policy in the Integrated Cache

You configure new policies to handle data that the built-in policies cannot process. You configure separate policies for caching, preventing caching from occurring, and for invalidating cached data. Following are the main components of a policy for integrated caching:

- Rule: A logical expression that evaluates an HTTP request or response.
- Action: You associate a policy with an action to determine what to do with a request or response that matches the policy rule.
- Content groups: You associate the policy with one or more content groups to identify where the action is to be performed.

To configure a policy for caching by using the NetScaler command line

At the NetScaler command prompt, type:

add cache policy <policyName> **-rule** <expression> **-action** CACHE|MAY_CACHE|NOCACHE|MAY_NOCACHE [**-storeInGroup** <contentGroupName>] [**-undefAction** NOCACHE|RESET]

Examples

```
add cache policy image_cache -rule "http.req.url.contains(\"jpg\") || http.req.url.contains(\"jpeg\")" -action
add cache policy bugReportPolicy -rule "http.req.url.query.contains(\"IssuePage\")" -action CACHE -storeInGr
add cache policy my_form_policy -rule "http.req.header(\"Host\")contains(\"my.company.com\") && http.req
add cache policy viewproducts_policy –rule "http.req.url.contains(\"viewproducts.aspx\")" –action CACHE -sto
```

To configure a policy for invalidation by using the NetScaler command line

At the NetScaler command prompt, type:

add cache policy <policyName> **-rule** <expression> **-action** INVAL [**-invalObjects** "<contentGroupName1>[,<selectorName1>"]. . .]] | [**-invalGroup** <contentGroupName1>[, <contentGroupName2>. . .]] [**-undefAction** NOCACHE|RESET]

Examples

```
add cache policy invalidation_events_policy -rule "http.req.header(\"Host\")contains(\"my.company.com\") &
add cache policy inval_all -rule "http.req.method.eq(\"POST\") && http.req.url.contains(\"jpeg\")" -action INV
add cache policy bugReportInvalidationPolicy -rule "http.req.url.query.contains(\"TransitionForm\")" -action
add cache policy editproducts_policy -rule "http.req.url.contains(\"editproducts.aspx\")" -action INVAL -inval
```

Parameters for configuring policies

policyName

The name that you want to assign to the policy:

rule

A logical expression, as described in Configuring Expressions for Caching Policies and Selectors.

action

The action to take: CACHE, MAY_CACHE, NOCACHE, MAY_NOCACHE, or INVAL.

storeInGroup

Only used if the action is CACHE, MAY_CACHE, NOCACHE, or MAY_NOCACHE.

The name of a content group where the response is stored. This is optional if you do not want to use the Default content group. For more information, see Configuring Selectors and Basic Content Groups.

undefAction

The action to take if the policy cannot be evaluated. The Global-Undefined-Result-Action indicates the action (NOCACHE or RESET) that is configured in the global settings for the integrated cache. You can also override the global action by selecting NOCACHE or RESET.

invalGroups

Only used if the action is INVAL.

Indicates the content groups to be invalidated. Maximum length of the list of groups is 511.

invalObjects

Only used if the action is INVAL. This parameter invalidates objects in the content group that match the specified invalidation selector.

Maximum Length: 1087.

Note: After configuring the policy you must bind the policy, as described in Globally Binding an Integrated Caching Policy.

To configure a policy for caching or invalidation by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Policies**.
2. In the details pane, do one of the following:
   - To create a new policy for caching or invalidation, click **Add**.
   - To modify an existing policy for caching or invalidation, select the policy, and then click **Open**.
3. In the **Create Cache Policy** or **Configure Cache Policy** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring policies” as shown:
   - **Name***—name
   - **Action***—action
   - **Store in Group**—storeInGroup
   - **Undefined-Result Action**—undefAction
   - **Expression**—rule
   - **Invalidate all objects in the following groups**—invalGroups
   - **Invalidate selected objects in the following parameterized groups**—invalObjects

   * A required parameter.
4. Click **Create**, and then click **Close**.

Globally Binding an Integrated Caching Policy

When you globally bind a policy, it is available to all virtual servers on the NetScaler appliance.

To bind an integrated caching policy globally by using the NetScaler command line

At the NetScaler command prompt, type:

bind cache global <policyName> **-priority** <positiveInteger> [**-type** REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] [**-gotoPriorityExpression** <expression>] [**-invoke** <labelType> <labelName>]

Example

bind cache global myCachePolicy -priority 100 -type req_default

Note that the type argument is optional for globally bound policies, to maintain backward compatibility with policies that you defined using earlier versions of the NetScaler appliance. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. If the rule contains both request time and response time parameters, it is bound to RES_DEFAULT. Following is an example of a binding that omits the type.

bind cache global myCache Policy 200

To bind an integrated caching policy globally by using the configuration utility

1. In the navigation pane, click **Integrated Caching**.
2. In the details pane, click **Cache policy manager**.
3. In the **Cache Policy Manager** dialog box, select a **Request** or **Response** bind point, and then select a second level of binding of either Override Global or Default Global. A list of policies appears. These are policies that are bound to this bind point.
4. Click **Insert Policy** and do one of the following:
   - To configure a new policy, click **New Policy** and configure the new policy as described in Configuring a Policy in the Integrated Cache.
   - To bind an existing policy, click the name of the policy.
5. Drag and drop the policy to the position in the policy bank where you want it to be evaluated, or manually enter a priority level, as a positive integer, for this entry in the **Priority** field.
6. Optionally, to configure a Goto expression as described in the *Citrix NetScaler Policy Configuration and Reference Guide* at , double-click the field in the **Goto Expression** column, and enter valid priority level, the keywords **NEXT** or **END**, or an advanced expression. See Entries to Control Evaluation Flow in a Policy Bank for details.
7. Optionally, to invoke an external policy bank, click the field in the **Invoke Type** column, and select the type of policy bank that you are adding (a global label or a virtual server bank). In the **Invoke Name** field, enter the label or virtual server name. See Entries to Control Evaluation Flow in a Policy Bank for details.
8. Click **Apply Changes**.

Binding an Integrated Caching Policy to a Virtual Server

When you bind a policy to a virtual server, it is available only to requests and responses that match the policy and that flow through the relevant virtual server.

When using the configuration utility, you can bind the policy using the configuration dialog box for the virtual server. This enables you to view all of the policies from all NetScaler modules that are bound to this virtual server. You can also use the Policy Manager configuration dialog for the integrated cache. This enables you to view only the integrated caching policies that are bound to the virtual server.

To bind an integrated caching policy to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb|cs vserver <virtualServerName> **-policyName** <policyName> **-priority** <positiveInteger> **-type** (REQUEST|RESPONSE)

To bind an integrated caching policy to a virtual server by using the configuration utility (virtual server method)

1. In the navigation pane, expand the module where you want to bind the caching policy, and then click **Virtual Servers**. This must be a module in which you can configure virtual servers. The choices are **Load Balancing** or **Content Switching**.

2. In the details pane, double-click the name of the virtual server with which you want to associate a policy for Integrated Caching, and then click **Open**.

3. In the **Configure Virtual Server** dialog box, click the **Policies** tab. In the list of policies in this dialog box, Integrated Caching policies can be identified in the **Type** column by the label **Cache**. Click the **Active** check box for the Integrated Caching policy that you want to bind to this bind point.

4. To configure a priority for the policy, double-click the value in the **Priority** field and enter a new positive integer value. The lower the value, the earlier the policy is evaluated.

5. In the **Flow Type** field for this entry, select **Request** or **Response**.

6. Click **OK**.

To bind an integrated caching policy to a virtual server by using the configuration utility (Policy Manager method)

1. In the navigation pane, click **Integrated Caching**.
2. In the details pane, click **Cache policy manager**.
3. In the **Cache Policy Manager** dialog box, select a **Request** or **Response** bind point, select a second level of binding of either **LB Virtual Server** or **CS Virtual Server**, and then select the name of a virtual server. A list of policies appears. These are integrated caching policies that are bound to this virtual server.
4. Click **Insert Policy** and do one of the following:
   - To configure a new policy, click **New Policy** and configure the new policy as described in Configuring a Policy in the Integrated Cache.
   - To bind an existing policy, click the name of the policy.
5. Drag and drop the policy to the position in the policy bank where you want it to be evaluated, or manually enter a priority level, as a positive integer, for this entry in the **Priority** field.
6. Optionally, to configure a Goto expression as described in the *Citrix NetScaler Policy Configuration and Reference Guide* at , double-click the field in the **Goto Expression** column, and enter valid priority level, the keywords **NEXT** or **END**, or an advanced expression. See Entries to Control Evaluation Flow in a Policy Bank.
7. Optionally, to invoke an external policy bank, click the field in the **Invoke Type** column, and select the type of policy bank that you are adding (a global label or a virtual server bank). In the **Invoke Name** field, enter the label or virtual server name. See Entries to Control Evaluation Flow in a Policy Bank.
8. Click **Apply Changes**.

Example: Caching Compressed and Uncompressed Versions of a File

By default, a client that can handle compression can be served uncompressed responses or compressed responses in gzip, deflate, compress, and pack200-gzip format. If the client handles compression, an Accept-Encoding:compression format header is sent in the request. The compression type accepted by the client must match the compression type of the cached object. For example, a cached .gzip file cannot be served in response to a request with an Accept-Encoding:deflate header.

A client that cannot handle compression is served a cache miss if the cached response is compressed.

For dynamic caching, you need to configure two content groups, one for compressed data and one for uncompressed versions of the same data. The following is an example of configuring the selectors, content groups, and policies for serving uncompressed files from the cache to clients that cannot handle compression, and serving compressed versions of the same files to client that can handle compression.

```
add cache selector uncompressed_response_selector http.req.url "http.req.header(\"Host\")"
add cache contentGroup uncompressed_group -hitSelector uncompressed_responst_selector -invalSelector u
add cache policy cache_uncompressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\") && !HTTP.REQ.HEADER(\"Acc
bind cache global cache_uncompressed -priority 100 -gotoPriorityExpression END -type REQ_OVERRIDE
add cache selector compressed_response_selector HTTP.REQ.URL "HTTP.REQ.HEADER(\"Host\")" "HTTP.REQ.H
add cache contentGroup compressed_group -hitSelector compressed_response_selector
add cache policy cache_compressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\") && HTTP.REQ.HEADER(\"Accept
bind cache global cache_compressed -priority 200 -gotoPriorityExpression END -type REQ_OVERRIDE
```

Configuring a Policy Bank for Caching

All of the policies that are associated with a particular bind point are collectively known as a policy bank. In addition to configuring priority levels for policies in a bank, you can modify the order of evaluation order in a bank by configuring Goto expressions. You can further modify the evaluation order by invoking an external policy bank from within the current policy bank. You can also configure new policy banks, to which you assign your own labels. Because such policy banks are not bound to any point in the processing cycle, they can be invoked only from within other policy banks. For convenience, policy banks whose labels do not correspond to a built-in bind point are called policy labels.

As described in detail in the *Citrix NetScaler Policy Configuration and Reference Guide* at , in addition to controlling order of policy evaluation by binding the policy and assigning a priority level, you can establish the flow within a bank of policies by configuring a Goto expression. A Goto expression overrides the flow that is determined by the priority levels. You can also control the evaluation flow by invoking an external policy bank after evaluating an entry in the current bank. Evaluation always returns to the current bank after evaluation has completed for the external bank.

The following table summarizes the entries to control evaluation in a policy bank.

Table 1. Entries to Control Evaluation Flow in a Policy Bank

Attribute	Specifies
Name	The name of a policy, or, to invoke another policy bank without evaluating the policy, the keyword NOPOLICY. You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.
Priority	An integer. The lower the integer, the higher the priority.

Goto Expression	Determines the next policy or policy bank to evaluate. You can provide one of the following values: • **NEXT**: Go to the policy with the next higher priority. • **END**: Stop evaluation. • **USE_INVOCATION_RESULT**: Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT. • **Positive number**: Priority number of the next policy to be evaluated. • **Numeric expression**: Expression that produces the priority number of the next policy to be evaluated. The Goto can only proceed forward in a policy bank. Omitting the Goto expression is the same as specifying **END**.
Invocation Type	Designates a policy bank type. The value can be one of the following: • **Request Vserver**: Invokes request-time policies that are associated with a virtual server. • **Response Vserver**: Invokes response-time policies that are associated with a virtual server. • **Policy label**: Invokes another policy bank, as identified by thex policy label for the bank.
Invocation Name	Name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

The integrated cache has two built-in policy labels, and you can configure additional policy labels:

- **_reqBuiltInDefaults**: This policy label is invoked from the request-time default bind point.
- **_resBuiltInDefaults**: This policy label is invoked from the response-time default bind point.

Note: For information about creating policy labels, see Configuring a Policy Label in the Integrated Cache.

To invoke a policy label in a caching policy bank by using the NetScaler command line

At the NetScaler command prompt, type:

bind cache policylabel <policylabelName> <policyName> <priority>
[**-gotoPriorityExpression** <gotopriorityExpression>] [**-invoke** <labelType> <labelName>]

To invoke a policy label in a caching policy bank by using the configuration utility

1. In the navigation pane, click **Integrated Caching**.
2. In the details pane, click **Cache policy manager**.
3. In the **Cache Policy Manager** dialog box, select a **Request** or **Response** bind point, and then select a second level of binding of either **Override Global** or **Default Global**. A list of policies appears. These are policies that are bound to this bind point.
4. If you want to invoke a policy label without evaluating a policy, select **Insert Policy** and select the keyword **NOPOLICY**.

   If you want to invoke a policy label after processing a policy, skip this step.
5. To invoke an external policy bank, click the field in the **Invoke Type** column, and select the type of policy bank that you want to invoke at this point in the policy bank. This can be a global label or a virtual server bank. In the **Invoke Name** field, enter the label or virtual server name. See Entries to Control Evaluation Flow in a Policy Bank for details.
6. Click **Apply Changes**.

To invoke a caching policy label in a virtual server policy bank by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb|cs vserver <virtualServerName> **-policyName** <policyName>|<NOPOLICY-CACHE>
-priority <positiveInteger> **-gotoPriorityExpression** <expression> **-type** REQUEST|RESPONSE
-invoke <labelType> <labelName>

For more information, see Entries to Control Evaluation Flow in a Policy Bank.

To invoke a caching policy label in a virtual server policy bank by using the configuration utility

1. In the navigation pane, click **Load Balancing** or **Content Switching**, as appropriate, and then click **Virtual Servers**.

2. Double-click the virtual server where you want to configure the policy bank, and in the **Configure Virtual Server** dialog box, click the **Policies** tab.

3. If you are configuring an existing entry in this bank, skip this step. If you are adding a new policy to this policy bank, or you want to use the “dummy” NOPOLICY entry, click **Add Policy**, and do one of the following:

   - To configure a new policy, click **Cache** and configure the new policy as described in Configuring a Policy in the Integrated Cache.

   - To invoke a policy bank without processing a policy a rule, select the **NOPOLICY-CACHE** option.

   After configuring the new entry, it appears at the bottom of the list of entries with the name of the policy in the **Policy Name** field.

4. To bind the entry to this policy label, ensure the **Active** check box is selected.

5. Enter a priority level for this entry in the **Priority** field. The priority is a positive integer.

6. Optionally, to configure a Goto Expression, double-click the field in the **Goto Expression** column, and enter valid priority number, the keyword **NEXT** or **END**, or an advanced expression. For more information, see Entries to Control Evaluation Flow in a Policy Bank.

7. To invoke another policy bank, click the field in the **Invoke Type** column, and select the type of policy bank that you are adding (a global label or a virtual server bank). In the **Invoke Name** field enter the label or virtual server name. When you are done, click **OK**. For more information, see Entries to Control Evaluation Flow in a Policy Bank.

Configuring a Policy Label in the Integrated Cache

In addition to configuring policies in a policy bank for one of the built-in bind points or a virtual server, you can create caching policy labels and configure banks of policies for these new labels.

A policy label for the integrated cache can be invoked only from one of the bind points that you can view in the Policy Manager in the **Integrated Caching** details pane (request override, request default, response override, or response default) or the built-in policy labels **_reqBuiltinDefaults** and **_resBuiltinDefaults**. You can invoke a policy label any number of times unlike a policy, which can only be invoked once.

When you create a policy label, you specify whether it is invoked at request time or response time.

Note: You can use the NOPOLICY "dummy" policy to invoke any policy label from another policy bank. The NOPOLICY entry is a placeholder that does not process a rule.

To configure a policy label for caching by using the NetScaler command line

At the NetScaler command prompt, type the following command to create a policy label and verify the configuration:

- add cache policylabel <policyLabelName> **-evaluates** (REQ|RES)
- show cache policylabel <policyLabelName>

Invoke this policy label from a policy bank. For more information, see Configuring a Policy Bank for Caching.

To configure a policy label for caching by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Policy Labels**.
2. If you are configuring a new policy label, click **Add**. Then, in the **Create Cache Policy Label** dialog box, enter a name. In the **Evaluates** drop-down menu, select whether this is a **request-time (REQ)** or **response-time (RES)** policy label.

   If you are configuring an existing label, from the Policy Labels page, double-click the label.
3. To add a policy to this policy label, click **Insert Policy**.
4. Optionally, you can invoke other policy labels from this policy label, as described in Configuring a Policy Bank for Caching.
5. To ensure that the NetScaler appliance processes the policy label at the right time, you configure an invocation of this label in one of the banks that are associated with the built-in bind points, as described in Configuring a Policy Bank for Caching.

Unbinding and Deleting an Integrated Caching Policy and Policy Label

You can unbind a policy from a policy bank, and you can delete it. To delete the policy, you must first unbind it. You can also remove a policy label invocation and delete a policy label. To delete the policy label, you must first remove any invocations that you have configured for the label.

You cannot unbind or delete the labels for the built-in bind points (request default, request override, response default, and response override).

To unbind a global caching policy by using the NetScaler command line

At the NetScaler command prompt, type:

unbind cache global <policyName>

To unbind a virtual server-specific caching policy by using the NetScaler command line

At the NetScaler command prompt, type:

(unbind lb vserver|unbind cs vserver) <vserverName> **-policyName** <policyName> **-type** (REQUEST|RESPONSE)

To delete a caching policy by using the NetScaler command line

At the NetScaler command prompt, type:

rm cache policy <policyName>

To unbind a caching policy by using the configuration utility

1. In the navigation pane, click **Integrated Caching**.
2. In the details pane, click **Cache policy manager**.
3. In the **Cache Policy Manager** dialog box, select a **Request** or **Response** bind point, select a second level of binding of either **LB Virtual Server** or **CS Virtual Server**, and then select the name of a virtual server. A list of policies appears. These are integrated caching policies that are bound to this virtual server.
4. Select the caching policy you want to unbind, and then click **Unbind Policy**.
5. Click **Close**.

To delete a policy label invocation by using the configuration utility

1. In the navigation pane, click **Integrated Caching**.
2. In the details pane, click **Cache policy manager**.
3. In the **Cache Policy Manager** dialog box, select a **Request** or **Response** bind point, select a second level of binding of either **LB Virtual Server** or **CS Virtual Server**, and then select the name of a virtual server. A list of policies appears. These are integrated caching policies that are bound to this virtual server.
4. In the **Invoke** column for the policy label you want to invoke, click the drop-down list and clear the entry.
5. Click **Close**.

Configuring Expressions for Caching Policies and Selectors

A request-time expression examines data in request-time transaction, and a response-time expression examines data in a response-time transaction. In a policy for caching, if an expression matches data in a request or response, the Citrix® NetScaler® appliance takes the action associated with the policy. In a selector, request-time expressions are used to find matching responses that are stored in a content group.

Before configuring policies and selectors for the integrated cache, you need to know, at minimum, the host names, paths, and IP addresses that appear in HTTP request and response URLs. And you probably need to know the format of entire HTTP requests and responses. Programs such as Live HTTP Headers (http://livehttpheaders.mozdev.org/) or HTTPFox (https://addons.mozilla.org/en-US/firefox/addon/6647) can help you investigate the structure of the HTTP data that your organization works with.

Following is an example of an HTTP GET request for a stock quote program:

```
GET /quote.dll?page=dynamic&mode=data&mode=stock&symbol=CTXS&page=multi&selected=CTXS&random
Host: quotes.mystockquotes.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9) Gecko/2008052906 Firefox/3.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate,compress,pack200-gzip
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://quotes.mystockquotes.com/quote.dll?mode=stock&symbol=CTXS&page=multi&selected=CTX
Cookie: __qca=1210021679-72161677-10297606
```

When configuring an expression, note the following limitations:

Table 1. Restrictions on Request-Time and Response-Time Expressions

Expression Type	Restrictions
Request	Do not configure request-time expressions in a policy with a CACHE or NOCACHE action. Use MAY_CACHE or MAY_NOCACHE instead.

Response	Configure response-time expressions in caching policies only. • Selectors can use only request-time expressions. • Do not configure response-time expressions in a policy with an INVAL action. Do not configure response-time expressions in a policy with a CACHE action and a parameterized content group. Use the MAY_CACHE action.

Note: For a comprehensive discussion of advanced expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

Expression Syntax

Following are basic components of the syntax:

- Separate keywords with periods (.), as follows:

  ```
  http.req.url
  ```

- Enclose string values in parentheses and quotes, as follows:

  ```
  http.req.url.query.contains("this")
  ```

- When configuring an expression from the command line, you must escape internal quote marks (the quotes that delimit values in the expression, as opposed to the quotes that delimit the expression). One method is to use a slash, as followings:

  ```
  \"abc\"
  ```

Selector expressions are evaluated in order of appearance, and multiple expressions in a selector definition are joined by a logical AND. Unlike selector expressions, you can specify Boolean operators and modify the precedence in an advanced expression for a policy rule.

Configuring an Expression in a Caching Policy or a Selector

Note that on the command line, the syntax for a policy expression is somewhat different from a selector expression. For a comprehensive discussion of advanced expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

To configure a policy expression by using the NetScaler command line

1. Start the policy definition as described in Globally Binding an Integrated Caching Policy.
2. To configure the policy rule, delimit the entire rule in quotes, and delimit string values within the rule i

   The following is an example:

   "http.req.url.contains(\"jpg\")"
3. To add Boolean values, insert &&, ||, or ! operators.

   The following are examples:

   "http.req.url.contains(\"jpg\") || http.req.url.contains(\"jpeg\")"
   "http.req.url.query.contains(\"IssuePage\")"
   "http.req.header(\"Host\")contains(\"my.company.com\") && http.req.method.eq(\"GET\") && http.req.u
4. To configure an order of evaluation for the constituent parts of a compound

   "http.req.url.contains(\"jpg\") || (http.req.url.contains(\"jpeg\") && http.req.method.eq(\"GET\"))"

To configure a selector expression by using the NetScaler command line

1. Start the selector definition as described in About Content Groups.
2. To configure the selector expression, delimit the entire rule in quotes, and delimit string values within t quotes.

   The following is an example:

   "http.req.url.contains(\"jpg\")"
3. You cannot add Boolean values, insert &&, ||, or ! operators. Enter each expression element delimited i expressions in the definition are treated as a compound expression joined by logical ANDs.

   The following are examples:

   "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"BatchNum\")" "http.req.url.query.val

To configure a policy or selector expression by using the configuration utility

1. Start the policy or selector definition as described in To configure a policy for caching or invalidation by using the configuration utility or To configure a selector by using the configuration utility.

2. Click in the **Expression** field.

3. Click the **Prefix** icon (the house) and select the first expression prefix from the drop-down list. The options are **HTTP**, **SYS**, **CLIENT**, and **SERVER**. The next set of applicable options appears in a drop-down list.

4. Double-click the next option to select it, and then type a period (.). Again, a set of applicable options appear in another drop-down list.

5. Continue selecting options until an entry field (indicated by parentheses) appears. When you see an entry field, enter an appropriate value in the parentheses. For example, if you select **GT(int)** (greater-than, integer format), you specify an integer in the parentheses. Text strings are delimited by quote marks. The following is an example:

   ```
   HTTP.REQ.BODY(1000).BETWEEN("this","that")
   ```

6. To insert an operator between two parts of a compound expression, click the **Operators** icon (the sigma), and select the operator type. The following is an example of a configured expression with a Boolean OR (signaled by double vertical bars, ||):

   ```
   HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this","that")
   ```

7. To insert a named expression, click the down arrow next to the **Add** icon (the plus sign) and select a named expression. For more information about named expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

8. To configure an expression using drop-down menus, and to insert built-in expressions, click the **Add** icon (the plus sign). The **Add Expression** dialog box works in a similar way to the main dialog box, but it provides drop-down lists for selecting options, and it provides text fields for data entry instead of parentheses. This dialog box also provides a **Frequently Used Expressions** drop-down list that inserts commonly used expressions. When you are done adding the expression using this dialog box, click **OK**.

9. To test the expression, click the **Evaluate** icon (the check mark). In the **Advanced Expression Evaluator** dialog box, select the **Flow Type** that matches the expression. In the data field, paste the HTTP request or response that you hope to parse using the expression, and click **Evaluate**. Click **OK** to save your expression and close this dialog box.

10. When you are done, click **Create** and then click **Close**.

Displaying Cached Objects and Cache Statistics

You can view particular cached objects, and you can view summary statistics on cache hits, misses, and memory usage. The statistics provide insight on the amount of data that is being served from the cache, what items are responsible for the largest performance benefit, and what you can tune to improve cache performance.

Viewing Cached Objects

After enabling caching, you can view details for cached objects. For example, you can view the following items:

- Response sizes and header sizes
- Status codes
- Content groups
- ETag, Last-Modified, and Cache-Control headers
- Request URLs
- Hit parameters
- Destination IP addresses
- Request and response times

To view a list of cached objects by using the NetScaler command line

At the NetScaler command prompt, type:

show cache object

Table 1. Properties of Cached Objects

Properties	Specifies
Response size (bytes)	The size of the response header and body.
Response header size (bytes)	The size of the header portion of the response.
Response status code	The status code sent with the response.
ETag	The ETag header inserted in the response. Typically, this header indicates whether the response has changed recently.
Last-Modified	The Last-Modified header inserted in the response. This header indicates the date that the response was last changed.

Cache-Control	The Cache-Control header inserted in the response.
Date	The Date header that indicates when the response was sent.
Contentgroup	The content group where the response is stored.
Complex match	If this object was cached on the basis of parameterized values, this field value is YES.
Host	The host specified in the URL that requested this response.
Host port	The listen port for the host specified in the URL that requested this response.
URL	The URL issued for the stored response.
Destination IP	The IP address of the server from which this response was fetched.
Destination port	The listen port for the destination server.
Hit parameters	If the content group that stores the response uses hit parameters, they are listed in this field.
Hit selector	If this content group uses a hit selector, it is listed in this field.
Inval selector	If this content group uses an invalidation selector, it is listed in this field.
Selector Expressions	If this content group uses a selector, this field displays the expression that defines the selection rule.
Request time	The time in milliseconds since the request was issued.
Response time	The time in milliseconds since the cache started to receive the response.
Age	Amount of time the object has been in the cache.
Expiry	Amount of time after which the object is marked as expired.
Flushed	Whether the response has been flushed after expiry.
Prefetch	If Prefetch has been configured for this content group, the amount of time before expiry during which the object is fetched from the origin. Prefetch does not apply to negative objects (for example, 404 "object not found" responses).

Current readers	Approximately the current number of hits being served. When a response with a Content-Length header object is being downloaded, the current misses and the current readers values are each typically 1. When a chunked response object is being downloaded, the current misses value is typically 1, but the current readers value is typically 0, because the chunked response that is served to the client does not come from the integrated caching buffers.
Current misses	The current number of requests that resulted in a cache miss and fetching from the origin server. This value is typically 0 or 1. If Poll Every Time is enabled for a content group, the count can be greater than 1.
Hits	The number of cache hits for this object.
Misses	The number of cache misses for this object.
Compression format	The type of compression applied to this object. Compression formats include gzip, deflate, compress, and pack200-gzip.
HTTP version in response	The version of HTTP that was used to send the response.
Weak etag present in response	Strong etag headers change if the bits of an entity change. Strong headers are based on the octet values of an object. Weak etag headers change if the meaning of an entity changes. Weak etag values are based on semantic identity. Weak etags values start with a "W."
Negative marker cell	A marker object is cacheable, but it does not yet meet all the criteria for being cached. For example, the object may exceed the maximum response size for the content group. A marker cell is created for objects of this type. The next time a user sends a request for this object, a cache miss is served.
Reason marker created	The reason a marker cell was created (for example, “Waiting for minhit,” “Content-length response data is not in group size limit”).
Auto poll every time	If the integrated cache receives an already expired 200 OK response with validators (either the Last-Modified or the ETag response headers) it stores the response and marks it as Auto-PET (automatically poll every time).

NetScaler Etag inserted in response	A variation of the ETag header generated by the NetScaler appliance. A value of YES appears if the NetScaler inserts an Etag in the response.
Full response present in cache	Indicates whether this is a complete response.
Destination IP verified by DNS	Indicates whether DNS resolution was performed when storing the object.
Object stored through a cache forward proxy	Indicates whether this response was stored due to a forward proxy that is configured in the integrated cache.
Object is a Delta basefile	A response that is delta-compressed.
Waiting for minhits	Indicates whether this content group requires a minimum number of origin server hits before caching a response.
Minhit count	If this content group requires a minimum number of origin server hits before caching an object, this field displays a count of the number of hits received so far.
HTTP Request Method	The method, GET or POST, used in the request that obtained this object.
Stored by policy	The name of the caching policy that caused this object to be stored. A value of NOT AVAILABLE indicates that the policy has been deactivated or deleted. A value of NONE indicates that the object did not match a visible policy, but was stored according to internal criteria for caching.
Application Firewall metadata exists	This parameter is used when the Application Firewall and the integrated cache are both enabled. The Application Firewall analyzes the contents of a response page, stores its metadata (for example, URLs and forms contained in page), and exports the metadata with the response to the cache. The cache stores the page and the metadata, and when the cache serves the page, it sends the metadata back to the request's session.
HTTP callout object, name, type, response	These cells indicate whether this data was stored as a result of an HTTP Callout expression, and provide information about various aspects of the callout and the corresponding response. For more information about HTTP callouts, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

To view cached objects by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Cache Objects**.

   By default, all cached responses are displayed. If you have not yet configured any content groups, all of the responses are in the Default group.

2. To filter the list by content group, click the **Content Group Name** drop-down menu, and then select the name of a content group.

3. To filter out marker objects and not-ready objects, click the check box for **Ignore Marker Objects** and leave the check box for **Include Not Ready Objects blank**.

   These settings represent objects that can be served from the cache, pending additional information. Marker objects are responses that have not yet reached a minimum number of hits before being cached. Not-ready objects have not yet received response headers.

4. Click **Go**.

   To view details for a cached object, click it, and then click **Details**.

5. To save a locator number for later use, right-click the row that contains the object and its locator number, select **Copy**, and then paste the information into a document.

Finding Particular Cached Responses

You can find individual items in the cache based on search criteria. There are different methods for finding cached items, depending on whether the content group that contains the data uses hit and invalidation selectors, as follows:

- If the content group uses selectors, you can only conduct the search using the Locator ID for the cached item.
- If the content group does not use selectors, you conduct the search using criteria such as URL, host, content group name, and so on.

When searching for a cached response, you can locate some items by URL and host. If the response is in a content group that uses a selector, you can find it only by using a Locator number (for example, 0x00000000ad7af00000050). To save a Locator number for later use, right-click the entry and select **Copy**. For more information about selectors, see Configuring Selectors and Basic Content Groups.

To display cached responses in content groups that do not have a selector by using the NetScaler command line

At the NetScaler command prompt, type:

show cache object [**-locator** <positiveInteger>] | [(**-url** <URL> (**-host** <hostName> [**-port** <port>] [**-groupName** <contentGroupName>] [**-httpMethod** GET | POST])) | | **-group** <contentGroupName> | **-ignoreMarkerObject** (ON | OFF) | **-includeNotReadyObjects** (ON | OFF)]

To display cached responses in content groups that have a selector by using the NetScaler command line

At the NetScaler command prompt, type:

show cache object **-locator** <locatorString> **MarkerObjects** (ON | OFF) | **-includeNotReadyObjects** (ON | OFF)]

Parameters for displaying cached items in a content group

url

The URL of the item in the cache. Maximum Length: 1023

host, port

The name of the host where the cached data was retrieved. Maximum Length: 511

httpMethod

The HTTP request method that caused the data to be stored.

Possible values: GET, POST

Default value: GET

groupName

The name of the content group in which a particular cached item is present.

Maximum length: 31

locator

Unique identification number of the cached response, expressed as a positive hexadecimal number.

Maximum: 1.84467440737096e+19.

group

The name of the content group whose objects you want to list.

Maximum Length: 31

ignoreMarker Objects

Marker objects are created when a response exceeds the maximum or minimum response size for the content group, or has not yet received the minimum number of hits for the content group.

Possible values: ON, OFF

ignoreNotReady Objects

Responses that have not yet reached a minimum number of hits before being cached.

Possible values: ON, OFF

To display cached responses in content groups that do not have a selector by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Cache Objects**.

   If you have not yet configured any content groups, all of the objects are in the **Default** group.

2. At the top of the details pane, click **Find**.

3. Enter search criteria, as follows:

   - In the **Search In** drop-down list, select a search filter, for example, **URL**. Note that if you are searching for items in a content group that uses a selector, select the **Locator** option.

   - In the **Criterion** drop-down list, select the search method, for example, **Contains**.

   - In **Look for**, enter the text that is to be matched, for example, **www.myurl.com**.

   - Optionally, restrict the search according to a particular content group, and exclude not ready and marker objects.

4. Click **Find Now**.

To display cached responses in content groups that have a selector by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Cache Objects**.

2. In the details pane, click **Look Up**.

3. In the **Look Up Cache Object** dialog box, do one of the following:

   - To find an object that resides in a content group that does not use a selector, click **Only static and parameterized groups** and enter selection criteria.

     The URL and host IP address are required. For example, you could enter /manual/images/sub.gif as the URL and 10.102.91.160 as the host. The content group name is also required if the content group contains parameterized objects.

   - To find an object that resides in a content group that uses a selector, click **All groups including selector-based** and, in **Locator**, enter the locator of the object that you want to view.

4. Click **View Details**.

5. In the **Details** dialog box, click any parameter to display its value in the **Value** box.

Viewing Cache Statistics

The following table summarizes the detailed cache statistics that you can view.

Table 1. Integrated Cache Statistics

Counter	Specifies
Hits	Responses that are found in and served from the integrated cache. Includes static objects such as image files, pages with status codes 200, 203, 300, 301, 302, 304, 307, 403, 404, 410, and responses that match a user-defined policy with a CACHE action..
Misses	Intercepted HTTP requests where the response was ultimately fetched from origin server.
Requests	Total cache hits plus total cache misses.
Non-304 hits	If the user requests an item more than once, and the item in the cache is unchanged since the last time the NetScaler appliance served it, the NetScaler appliance serves a 304 response instead of the cached object. This statistic indicates how many items the NetScaler appliance served from the cache, excluding 304 responses.
304 hits	Number of 304 (object not modified) responses the NetScaler appliance served from the cache.
304 hit ratio (%)	Percentage of 304 responses that the NetScaler appliance served, relative to other responses.
Hit ratio (%)	Percentage of responses that the NetScaler appliance served from the cache (cache hits) relative to responses that could not be served from the cache.
Origin bandwidth saved (%)	An estimate of the processing capacity that the NetScaler appliance saved on the origin server due to serving responses from the cache.
Bytes served by the NetScaler	Total number of bytes that the NetScaler appliance served from the origin server and the cache.
Bytes served by cache	Total number of bytes that the NetScaler appliance served from the cache.

Byte hit ratio(%)	Percentage of data that the NetScaler appliance served from the cache, relative to all of the data in all served responses.
Compressed bytes from cache	Amount of data, in bytes, that the NetScaler appliance served in compressed form.
Storable misses	If the NetScaler appliance does not find a requested object in the cache, it fetches the object from the origin server. This is known as a cache miss. A storable cache miss can be stored in the cache.
Non-storable misses	A non-storable cache miss cannot be stored in the cache.
Misses	All cache misses.
Revalidations	Max-Age setting in a Cache-Control header determines, in number of seconds, when an intervening cache must revalidate the content with the integrated cache before serving it to the user. For more information, see Inserting a Cache-Control Header.
Successful revalidations	Number of re-validations that have been performed. For more information, see Inserting a Cache-Control Header.
Conversions to conditional req	A user-agent request for a cached PET object is always converted to a conditional request and sent to the origin server. For more information, see Polling the Origin Server Every Time a Request Is Received.
Storable miss ratio (%)	Storable cache misses as a percentage of non-storable cache misses.
Successful reval ratio (%)	Successful revalidations as a percentage of all revalidation attempts. For more information, see Inserting a Cache-Control Header.
Expire at last byte	Number of times that the cache expired content immediately after receiving the last body byte. Only applicable to positive responses, as described in the table Cache Hits and Misses. For more information, see Example of Performance Optimization.

Flashcache misses	If you enable Flash Cache, the cache allows only one request to reach the server, eliminating flash crowds. This statistic indicates the number of Flash Cache requests that were cache misses. For more information, Queuing Requests to the Cache.
Flashcache hits	Number of Flash Cache requests that were cache hits. For more information, see Queuing Requests to the Cache.
Parametrized inval requests	Requests that match a policy with an invalidation (INVAL) action and a content group that uses an invalidation selector or parameters to selectively expire cached objects in the group.
Full inval requests	Requests that match an invalidation policy where the invalGroups parameter is configured and expires one or more content groups.
Inval requests	Requests that match an invalidation policy and result in expiration of specific cached responses or entire content groups.
Parameterized requests	Number of cache requests that were processed using a policy with a parameterized content group.
Parameterized non-304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where full cached response was found, and the response was not a 304 (object not updated) response.
Parameterized 304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found, and the object was a 304 (object not updated) response.
Total parameterized hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found.
Parameterized 304 hit ratio (%)	Percentage of 304 (object not updated) responses that were found using a parameterized policy, relative to all cache hits.
Poll every time requests	If Poll Every Time is enabled, the NetScaler appliance always consults the origin server before serving a stored object. For more information, see Polling the Origin Server Every Time a Request Is Received.

Poll every time hits	Number of times a cache hit was found using the Poll Every Time method. For more information, see Polling the Origin Server Every Time a Request Is Received.
Poll every time hit ratio (%)	Percentage of cache hits using the Poll Every Time method, relative to all searches for cached objects using Poll Every Time. For more information, see Polling the Origin Server Every Time a Request Is Received.
Maximum memory (KB)	Maximum amount of memory in the NetScaler appliance that is allocated to the cache. For more information, see Configuring Global Attributes for Caching.
Maximum memory active value (KB)	Maximum amount of memory (active value) that will be set after the memory is actually allocated to the cache. For more information, see *"Integrated Caching Configuration Scenarios in Release 9.2"* at http://support.citrix.com/article/CTX124553.
Utilized memory (KB)	Amount of memory that is actually being used.
Memory allocation failures	Number of failed attempts to utilize memory for the purpose of storing a response in the cache.
Largest response so far	Largest response in bytes found in either the cache or the origin server and sent to the client.
Cached objects	Number of objects in the cache, including responses that have not yet been fully downloaded and responses that have been expired but not yet flushed.
Marker objects	Marker objects are created when a response exceeds the maximum or minimum response size for the content group, or has not yet received the minimum number of hits for the content group.
Hits being served	Number of hits that have been served from the cache.
Misses being handled	Responses that were fetched from the origin server, stored in the cache, and then served. Should approximate the number for storable misses. Does not include non-storable misses.

To view summary cache statistics by using the NetScaler command line

At the NetScaler command prompt, type:

stat cache

To view specific cache statistics by using the NetScaler command line

At the NetScaler command prompt, type the following command:

stat cache **-detail** [**-fullValues**] [**-ntimes** <positiveInteger>] [**-logFile** <inputFilename>]

Parameters for viewing cache statistics

detail

Detailed statistics. Without this argument, summary statistics are provided

fullValues

Numbers and strings in their full form. Without this option, long strings are shortened and large numbers are abbreviated.

ntimes

The historic interval, in increments of seven seconds, to use for collecting the statistics. The default value is 1.

logFile

The name of the log file to be used as input.

To view summary cache statistics by using the configuration utility

1. In the details pane, click the **Monitoring** tab at the top of the page.
2. Scroll down to the **Integrated Caching** section of the window.
3. To see detailed statistics, click **More**.

To view specific cache statistics by using the configuration utility

1. In the navigation pane, click the **Reporting** tab at the top of the page.
2. Under **Built-In Reports**, expand **Integrated Cache**, and then click the report with the statistics you want to view.
3. To save the report as a template, click **Save As** and name the report. The saved report appears under **Custom Reports**.

Improving Cache Performance

You can improve the performance of integrated cache, including handling simultaneous requests for the same cached data, avoiding delays that are associated with refreshing cached responses from the origin server, and ensuring that a response is requested often enough to be worth caching.

Reducing Flash Crowds

Flash crowds occur when many users simultaneously request the same data. All of the requests in a flash crowd can become cache misses if you configured the cache to serve hits only after the entire object is downloaded.

The following techniques can reduce or eliminate flash crowds:

- **PREFETCH**: Refreshes a positive response before it expires to ensure that it never becomes stale or inactive.

  For more information, see Refreshing a Response Prior to Expiration.

- **Cache buffering**: Starts serving a response to multiple clients as soon as it receives the response header from the origin server, rather than waiting for the entire response to be downloaded.

  The only limit on the number of clients that can download a response simultaneously is the available system resources.

  The Citrix® NetScaler® appliance downloads and serves responses even if the client that initiated the download halts before the download is complete. If the size of the response exceeds the cache size or if the response is chunked, the cache stops storing the response, but service to the clients is not disrupted.

- **Flash Cache**: Flash Cache queues requests to the cache, and allows only one request to reach the server at a time.

  For more information, see Queuing Requests to the Cache.

Refreshing a Response Prior to Expiration

To ensure that a cached response is fresh whenever it is needed, the PREFETCH option refreshes a response before its calculated expiration time. The prefetch interval is calculated after receiving the first client request. From that point onward, the NetScaler appliance refreshes the cached response at a time interval that you configure in the PREFETCH parameter.

This setting is useful for data that is updated frequently between requests. It does not apply to negative responses (for example, 404 messages).

To configure prefetch for a content group by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <contentGroupName> **-prefetch** YES [**-prefetchPeriod** <seconds> | **-prefetchPeriodMilliSec** <milliseconds>] [**-prefetchMaxPending** <positiveInteger>]

Parameters for configuring prefetch for a content group

prefetch

The option to refresh an object immediately before it goes stale. Possible values: YES, NO. Default value: YES.

prefetchPeriod

The duration in seconds of the period during which prefetch should be attempted, immediately before the object's calculated expiry time.

prefetchPeriodMilliSec

The duration in milliseconds of the period during which prefetch should be attempted, immediately before the calculated expiry time.

prefetchMaxPending

The maximum number of outstanding prefetches on the content group.

To configure prefetch for a content group by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to set prefetch, and then click **Open**.
3. In the **Configure Cache Content Group** dialog box, on the **Others** tab, under **Flash Crowd and Prefetch**, select the **Prefetch** check box.
4. In the **Interval** text box, enter a value to define an interval that a prefetch can be attempted. This should be shorter than the content group's calculated expiration time. Or, accept the default (heuristic prefetch interval). Note that the value can be in seconds or, if relative expiry is specified in milliseconds on the **Expiry Method** tab, you can select a value in milliseconds.
5. In the **Maximum number of pending prefetches** text box, enter a value for the maximum number of prefetches that can be queued for the content group. The minimum value is 0, and the maximum value is 4294967295.
6. Click **OK**.

Queuing Requests to the Cache

The Flash Cache option queues requests that arrive simultaneously (a flash crowd), retrieves the response, and distributes it to all the clients whose requests are in the queue. If, during this process, the response becomes non-cacheable, the NetScaler appliance stops serving the response from the cache and instead serves the origin server's response to the queued clients. If the response is not available, the clients receive an error message.

Flash Cache is disabled by default. You cannot enable Poll Every Time (PET) and Flash Cache on the same content group.

One disadvantage of Flash Cache is if the server replies with an error (for example, a 404 that is quickly remedied), the error is fanned out to the waiting clients.

Note: If Flash Cache is enabled, in some situations the NetScaler appliance is unable to correctly match the Accept-Encoding header in the client request with the Content-Encoding header in the response. The NetScaler appliance can assume that these headers match and mistakenly serve a hit. As a work-around, you can configure Integrated Caching policies to disallow serving hits to clients that do not have an appropriate Accept-Encoding header.

To enable Flash Cache by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <contentGroupName> **-flashcache** yes

To enable Flash Cache by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to set the flash cache option, and then click **Open**.
3. In the **Configure Cache Content Group** dialog box, on the **Others** tab, under **Flash Crowd and Prefetch**, select the **Flash Cache** check box, and then click **OK**.

Caching a Response after a Client Halts a Download

You can set the Quick Abort parameter to continue caching a response, even if the client halts a request before the response is in the cache.

If the downloaded response size is less than or equal to the Quick Abort size, the NetScaler appliance stops downloading the response. If you set the Quick Abort parameter to 0, all downloads are halted. The default value is 4194303, minimum value is 0, and maximum value is 4194303.

To configure quick abort size by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <contentGroupName> **-quickAbortSize** <integerInKBytes>

To configure quick abort size by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to set the quick abort size, and then click **Add**.
3. In the **Configure Cache Content Group** dialog box, on the **Memory** tab, in the **Quick Abort: Continue caching if more than** text box, type an integer value, calculated in kilobytes (KB), and then click **OK**.

Setting a Minimum Number of Server Hits Prior to Caching

You can configure the minimum number of times that a response must be found on the origin server before it can be cached. You should consider increasing the minimum hits if the cache memory fills up quickly and has a lower-than-expected hit ratio. The default value is 0, the minimum value is 0, and the maximum value is 2147483647.

The default value for the minimum number of hits is 0. This value caches the response after the first request.

To configure the minimum number of hits that are required before caching by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <contentGroupName> **-minhits** <positiveInteger>

To configure the minimum number of hits that are required before caching by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to set the minimum hit count, and then click **Open**.
3. On the **Memory** tab, in the **Do not cache, if hits are less than** text box, type the value you want to set, and then click **OK**.

Example of Performance Optimization

In this example, a client accesses a stock quote. Stock quotes are highly dynamic. You configure the integrated cache to serve the same stock quote to concurrent clients without sending multiple requests to the origin server. The stock quote expires after it is downloaded to all of the clients, and the next request for a quote for the same stock is fetched from the origin server. This ensures that the quote is always up to date.

The following task overview describes the steps to configure the cache for the stock quote application.

Task overview: Configuring caching for a stock quote application

1. Create a content group for stock quotes.

   For more information, see About Content Groups.

   Configure the following for this content group:

   - On the **Expiry Method** tab, select the **Expire after complete response received** check box.
   - On the **Others** tab, select the **Flash Cache** check box, and click **Create**.
2. Add a cache policy to cache the stock quotes.

   For more information, see Configuring a Policy in the Integrated Cache.

   Configure the following for the policy:

   - In the **Action** and **Store in Group** lists, select **CACHE** and select the group that you defined in the previous step.
   - Click **Add**, and in the **Add Expression** dialog box configure an expression that identifies stock quote requests, for example:

     ```
     http.req.url.contains("cgi-bin/stock-quote.pl")
     ```
3. Activate the policy.

   For more information, see Globally Binding an Integrated Caching Policy. In this example, you bind this policy to request-time override processing and set the priority to a low value.

Configuring Cookies, Headers, and Polling

This section describes the procedures to configure how the cache manages cookies, HTTP headers, and origin server polling, including modifying default behavior that causes the cache to diverge from documented standards, overriding HTTP headers that might cause cacheable content to not be stored in the cache, and configuring the cache to always poll the origin for updated content under specialized circumstances.

Divergence of Cache Behavior from the Standards

By default, the integrated cache conforms to the following standards:

- RFC 2616, “Hypertext Transfer Protocol HTTP/1.1”
- The caching behaviors described in RFC 2617, “HTTP Authentication: Basic and Digest Access Authentication”
- The caching behavior described in RFC 2965, “HTTP State Management Mechanism”

The built-in policies and the Default content group attributes ensure conformance with most of these standards.

The default integrated cache behavior diverges from the specifications as follows:

- There is limited support for the Vary header.

  By default, any response containing a Vary header is considered to be non-cacheable unless it is compressed. A compressed response contains `Content-Encoding: gzip`, `Content-Encoding: deflate`, or `Content-Encoding: pack200-gzip` and is cacheable even if it contains the `Vary: Accept-Encoding` header.

- The integrated cache ignores the values of the headers `Cache-Control: no-cache` and `Cache-Control: private`.

  For example, a response that contains `Cache-Control: no-cache=”Set-Cookie”` is treated as if the response contained `Cache-Control: no-cache`. By default, the response is not cached.

- An image (Content-Type = image/*) is always considered cacheable even if an image response contains Set-Cookie or Set-Cookie2 headers, or if an image request contains a Cookie header.

  The integrated cache removes Set-Cookie and Set-Cookie2 headers from a response before caching it. This diverges from RFC 2965. You can configure RFC-compliant behavior as follows:

  ```
  add cache policy rfc_compliant_images_policy -rule "http.res.header.set-cookie2.exists || http.res.heac
  bind cache global rfc_compliant_images_policy -priority 100 -type REQ_OVERRIDE
  ```

- The following Cache-Control headers in a request force an RFC-compliant cache to reload a cached response from the origin server:

  ```
  Cache-control: max-age=0
  ```

Cache-control: no-cache

To guard against Denial of Service attacks, this behavior is not the default. For more information, see Inserting a Cache-Control Header.

- By default, the caching module considers a response to be cacheable unless a response header states otherwise.

  To make this behavior RFC 2616 compliant, set **-weakPosRelExpiry** and **-weakNegResExpiry** to 0 for all content groups.

Removing Cookies from a Response

Cookies are often personalized for a user, and typically should not be cached. The Remove Response Cookies parameter removes Set-Cookie and Set-Cookie2 headers before caching a response. By default, the Remove Response Cookies option for a content group prevents caching of responses with Set-Cookie or Set-Cookie2 headers.

Note that when images are cached, the built-in behavior is to remove the Set-Cookie and Set-Cookie2 headers before caching, no matter how the content group is configured.

Note: Citrix recommends that you accept the default Remove Response Cookies for every content group that stores embedded responses, for example, images.

To configure Remove Response Cookies for a content group by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <contentGroupName> **-removeCookies** YES

To configure Remove Response Cookies for a content group by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to enable or disable the remove response cookies option, and then click **Open**.
3. In the **Configure Cache Content Group**, on the **Others** tab, under **Settings**, select or clear the **Remove response cookies check** box, and then click **OK**.

Inserting HTTP Headers at Response Time

The integrated cache can insert HTTP headers in responses that result from cache hits. The Citrix® NetScaler® appliance does not alter headers in responses that result from cache misses.

The following table describes headers that you can insert in a response.

Table 1. Different HTTP Headers You Can Insert in a Response That Is Served from the Cache

Header	Specifies
Age	Provides the age of the response in seconds, calculated from the time the response was generated at the origin server. By default, the cache inserts an Age header for every response that is served from the cache.
Via	Lists protocols and recipients between the start and end points for a request or a response. The NetScaler appliance inserts a Via header in every response that it serves from the cache. The default value of the inserted header is "NS-CACHE-9.2:last octet of the NetScaler IP address." For more information, see Configuring Global Attributes for Caching.
ETag	The cache supports response validation using Last-Modified and ETag headers to determine if a response is stale. The cache inserts an ETag in a response only if it caches the response and the origin server has not inserted its own ETag header. The ETag value is an arbitrary unique number. The ETag value for a response changes if it is refreshed from the origin server, but it stays the same if the server sends a 304 (object not updated) response. Origin servers typically do not generate validators for dynamic content because dynamic content is considered non-cacheable. You can override this behavior. With ETag header insertion, the cache is permitted to not serve full responses. Instead, the user agent is required to cache the dynamic response sent by the integrated cache the first time. To force a user agent to cache a response, you configure the integrated cache to insert an ETag header and replace the origin-provided Cache-Control header.

Cache-Control	The NetScaler appliance typically does not modify cacheability headers in responses that is serves from the origin server. If the origin server sends a response that is labeled as non-cacheable, the client treats the response as non-cacheable even if the NetScaler appliance caches the response. To cache dynamic responses in a user agent, you can replace Cache-Control headers from the origin server. This applies only to user agents and other intervening caches. They do not affect the integrated cache. For more information, see Inserting a Cache-Control Header.

Inserting an Age, Via, or ETag Header

The following procedures describe how to insert Age, Via, and ETag headers.

To insert an Age, Via, or Etag header by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <name> **-insertVia** YES **-insertAge** YES **-insertETag** YES

To insert an Age, Via, or Etag header by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to insert Age, Via, or Etag headers, and then click **Open**.
3. In the **Configure Cache Content Group** dialog box, on the **Others** tab, under **HTTP Header Insertions**, select or clear the **Via, Age, or ETag** check boxes, as needed. The values for the other header types are calculated automatically. Note that you configure the Via value in the main settings for the cache.
4. Click **OK**.

Inserting a Cache-Control Header

When the integrated cache replaces a Cache-Control header that the origin server inserted, it also replaces the Expires header. The new Expires header contains an expiration time in the past. This ensures that HTTP/1.0 clients and caches (that do not understand the Cache-Control header) do not cache the content.

To insert a Cache-Control header by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <name> **-cacheControl** <value>

Parameters for inserting a Cache-Control header

private

The response is not cached in any other intervening cache.

public

Responses in a content group can be cached by the browser of the user and any intervening cache.

max-age

Number of seconds, after which an intervening cache must revalidate the content with the integrated cache before serving it to the user.

If you set the Max-Age value to 0, revalidation always occurs. Revalidation is efficient if the response contains a validator. For more information, see Inserting HTTP Headers at Response Time.

s-maxage

For a shared cache, this directive takes precedence over the max-age directive and the Expires header. Does not apply to a private cache.

no-store

Do not store the response or the corresponding request for it.

no-cache

An intervening cache must revalidate any subsequent requests for this object with the origin server. Typically not used with HTTP 1.0 caches.

no-transform

An intervening cache cannot alter the media type, as specified in Content-Encoding, Content-Range, and Content-Type headers.

must-revalidate

But if a response is stale, upon receipt of a new request for the object intervening caches must revalidate the response with the origin server, using the most recent request headers.

proxy-revalidate

Similar to must-revalidate, but does not apply to non-shared user agent caches. Note that authenticated responses also require the public directive.

To insert a Cache-Control header by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to configure cache-control, and then click **Open**.
3. In the **Configure Cache Content Group** dialog box, on the **Expiry Method** tab, clear the heuristic and default expiry settings if any are set, and in the **Expire content after** text box, type a value in seconds. For example, a value of 3600 expires cached responses after 3600 seconds (one hour).
4. On the **Others** tab, under **HTTP Header Insertions**, in the **Cache-Control** text box, type the header you want to insert, for example, **private**, **max-age=0**, or click **Configure** and set the Cache-Control directives that you want to insert in responses that are sent to user agents and intervening caches.

Ignoring Cache-Control and Pragma Headers in Requests

By default, the caching module processes Cache-Control and Pragma headers. The following tokens in Cache-Control headers are processed as described in RFC 2616.

- max-age
- max-stale
- only-if-cached
- no-cache

A Pragma: no-cache header in a request is treated in the same way as a Cache-Control: no-cache header.

If you configure the caching module to ignore Cache-Control and Pragma headers, a request that contains a Cache-Control: No-Cache header causes the NetScaler appliance to retrieve the response from the origin server, but the cached response is not updated. If the caching module processes Cache-Control and Pragma headers, the cached response is refreshed.

The following table summarizes the implications of various settings for these headers and the Ignore Browser's Reload Request setting.

Table 1. Outcome of Settings for Ignoring Reload Requests, Cache-Control, and Pragma Headers

Setting for Ignore Cache-Control and Pragma Headers	Setting for Ignore Browser's Reload Request	Outcome
Yes	Yes or No	Ignore the Cache-Control and Pragma headers from the client, including the Cache-Control: no-cache directive.
No	Yes	The Cache-Control: no-cache header produces a cache miss, but a response that is already in the cache is not refreshed.

No	No	A request that contains a Cache-Control: no-cache header causes a cache miss and the stored response is refreshed.

To ignore Cache-Control and Pragma headers in a request by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <name> **-ignoreReqCachingHdrs** YES

To ignore browser reload requests by using the NetScaler command line

At the NetScaler command prompt, type:

set cache contentgroup <name> **-ignoreReloadReq** NO

Note that by default, the -ignoreReloadReq parameter is set to YES.

To ignore Cache-Control and Pragma headers in a request by using the configuration utility

1. In the navigation pane, expand **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click the content group for which you want to ignore cache-control and pragma headers, and then click **Open**.
3. In the **Configure Cache Content Group** dialog box, on the **Others** tab, under **Settings**, select the **Ignore Cache-control and Pragma Headers in Requests** check box, and then click **OK**.

Example of a Policy to Ignore Cache-Control Headers

In the following example, you configure a request-time override policy to cache responses that contain Content-type: image/* regardless of the Cache-Control header in the response.

To configure a request-time override policy to cache all responses with image/*

1. Flush the cache using the **Invalidate All** option.

   For more information, see Flushing Responses in a Content Group.

2. Configure a new cache policy, and direct the policy to a particular content group. For more information, see Configuring a Policy in the Integrated Cache.

3. Ensure the content group that the policy uses is configured to ignore Cache-Control headers, as described in Ignoring Cache-Control and Pragma Headers in Requests.

4. Bind the policy to the request-time override policy bank.

   For more information, see Globally Binding an Integrated Caching Policy.

Polling the Origin Server Every Time a Request Is Received

You can configure the NetScaler appliance to always consult the origin server before serving a stored response. This is known as Poll Every Time (PET). When the NetScaler appliance consults the origin server and the PET response has not expired, a full response from the origin server does not overwrite cached content. This property is useful when serving client-specific content.

After a PET response expires, the NetScaler appliance refreshes it when the first full response arrives from the origin server.

The Poll Every Time (PET) function works as follows:

- For a cached response that has validators in the form of an ETag or a Last-Modified header, if the response expires it is automatically marked PET and cached.
- You can configure PET for a content group.

  If you configure a content group as PET, every response in the content group is marked PET. The PET content group can store responses that do not have validators. Responses that are automatically marked PET are always expired. Responses that belong to a PET content group can expire after a delay, based on how you configure the content group.

Two types of requests are affected by polling:

- **Conditional Requests**: A client issues a conditional request to ensure that the response that it has is the most recent copy.

  A user-agent request for a cached PET response is always converted to a conditional request and sent to the origin server. A conditional request has validators in If-Modified-Since or If-None-Match headers. The If-Modified-Since header contains the time from the Last-Modified header. An If-None-Match header contains the response's ETag header value.

  If the client's copy of the response is fresh, the origin server replies with 304 Not Modified. If the copy is stale, a conditional response generates a 200 OK that contains the entire response.
- **Non-Conditional Requests**: A non-conditional request can only generate a 200 OK that contains the entire response.

The following table summarizes response types based on the origin server's response

Table 1. How Responses Are Affected by Poll Every Time

Origin Server Response	Action
Send the full response	The origin server sends the response as-is to the client. If the cached response has expired, it is refreshed.
304 Not Modified	The following header values in the 304 response are merged with the cached response and the cached response is served to the client: • Date • Expires • Age • Cache-Control header Max-Age and S-Maxage tokens
401 Unauthorized 400 Bad Request 405 Method Not Allowed 406 Not Acceptable 407 Proxy Authentication Required	The origin's response is served as-is to the client. The cached response is not changed.
Any other error response, for example, 404 Not Found	The origin's response is served as-is to the client. The cached response is removed.

Note: The Poll Every Time parameter treats the affected responses as non-storable.

To configure poll every time by using the NetScaler command line

At the NetScaler command prompt, type:

add cache contentgroup <contentGroupName> **-pollEveryTime** YES

To configure poll every time by using the configuration utility

1. In the navigation pane, click **Integrated Caching**, and then click **Content Groups**.
2. In the details pane, click a cache content group, and then click **Open**.
3. On the **Others** tab, click **Poll every time (validate cached content with origin for every request)**.

PET and Client-Specific Content

The PET function can ensure that content is customized for a client. For example, a Web site that serves content in multiple languages examines the Accept-Language request header to select the language for the content that it is serving. For a multi-language Web site where English is the predominant language, all English language content can be cached in a PET content group. This ensures that every request goes to the origin server to determine the language for the response. If the response is English, and the content has not changed, the origin server can serve a 304 Not Modified to the cache.

The following example shows commands to cache English responses in a PET content group, configure a named expression that identifies English responses in the cache, and configure a policy that uses this content group and named expression. Bold is used for emphasis:

```
add cache contentgroup EnglishLanguageGroup -pollEveryTime YES
add expression containsENExpression –rule "http.res.header(\"Content-Language\").contains(\"en\")"
add cache policy englishPolicy -rule containsENExpression -action CACHE -storeInGroup englishLanguageGrou
bind cache policy englishPolicy -priority 100 -precedeDefRules NO
```

PET and Authentication, Authorization, and Auditing

Outlook Web Access (OWA) is a good example of dynamically generated content that benefits from PET. All mail responses (*.EML objects) have an ETag validator that enables them to be stored as PET responses.

Every request for a mail response travels to the origin server, even if the response is cached. The origin server determines whether the requestor is authenticated and authorized. It also verifies that the response exists in the origin server. If all results are positive, the origin server sends a 304 Not Modified response.

Configuring the Integrated Cache as a Forward Proxy

The integrated cache can service as a forward proxy device that passes requests to other Citrix® NetScaler® appliances or to other types of cache servers. You configure the integrated cache as a forward proxy by identifying the IP addresses of the cache server or servers. After configuring the forward proxy, the NetScaler appliance sends requests that contain the configured IP address on to the cache server instead of involving the integrated cache.

To configure the NetScaler as a forward cache proxy by using the NetScaler command line

At the NetScaler command prompt, type:

add cache forwardProxy <IPAddress> <port>

To configure the NetScaler as a forward cache proxy by using the configuration utility

1. In the navigation pane, click **Integrated Caching**, and then click **Forward Proxy**.
2. In the details pane, click **Add**.
3. Enter the IP address and port of the cache server, and then click **Create**.
4. Enter a second IP address and port, or click **Close**.

Example of an Integrated Caching Configuration

The following task overview provides one method of setting up and testing an end-to-end configuration for integrated caching.

Task overview: an Integrated Caching configuration

1. Configure the load balancing virtual servers for your environment, and bind services to them.
2. Enable the integrated cache.
3. Check the expiration settings of the Default content group and ensure there is enough time for objects to remain in the cache.
4. Check the memory settings and ensure there is enough room to store cached objects.
5. Configure a caching policy that stores responses in the cache. The policy can be very general for ease of testing. For example, the following rule produces a hit for any GET request:

   ```
   http.req.method.eq(GET)
   ```

6. Open a browser and enter `http://your_load_balancing_virtual_ip_address/known_content_on_the_destination server`. For example, if you have a Web server that contains a top-level file named myfile.txt and a virtual server that services the Web server with an IP address of 22.222.22.22, you could enter the following URL:

   ```
   http://22.222.22.22/myfile.txt
   ```

7. In the configuration utility click **Cache Objects**, or at the command line enter `show cache object`. If the expected object does not appear in the cache, check for conflicting caching policies. You can also ping the named destination server where the object of interest resides to see whether the IP address is the same as the one that you configured as a service on your virtual server. Finally, you can look in the configuration utility under **Integrated Caching**, on the landing page for **Policies**, in the **Hits** column for the policy, to see if the Citrix® NetScaler® appliance did use the intended policy to evaluate the request for the cached object.

The following is an example of a test configuration. The service IP address corresponds to a valid destination on the internet, and the load balancing virtual server IP address corresponds to a valid IP address in your network. You would send a browser request to the IP address for the virtual server:

```
enable ns feature lb ic
add service myTestService 11.111.11.11 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO
add lb vserver myTestLBVserver HTTP 10.100.10.10 80 -persistenceType NONE -cltTimeout 180
bind lb vserver myTestLBVserver myTestService
add cache policy myCachePolicy -rule "http.req.method.eq(\"GET\")" -action cache
bind cache global myCachePolicy -priority 1 -type req_override
```

Default Settings for the Integarated Cache

The Citrix® NetScaler® integrated cache feature provides built-in policies with default settings as well as initial settings for the Default content group. The information in this section defines the parameters for the built-in policies and Default content group.

Default Caching Policies

The integrated cache has built-in policies. The NetScaler appliance evaluates the policies in a particular order, as discussed in the following sections.

You can override these built-in policies with a user-defined policy that is bound to a request-time override or response-time override policy bank.

Note that if you configured policies prior to release 9.0 and specified the -precedeDefRules parameter when binding the policies, they are automatically assigned to override-time bind points during migration.

Viewing the Default Policies

The built-in policy names start with an underscore (_). You can view the built-in policies from the command line and the administrative console using the show cache policy command.

Default Request Policies

You can override the following built-in request time policies by configuring new policies and binding them to the request-time override processing point. In the following policies, note that the MAY_NOCACHE action stipulates that the transaction is cached only when there is a user-configured or built-in CACHE directive at response time.

The following policies are bound to the **_reqBuiltinDefaults** policy label. They are listed in priority order.

1. Do not cache a response for a request that uses any method other than GET.

   The policy name is **_nonGetRequest**. The following is the policy rule:

   ```
   !HTTP.REQ.METHOD.SET_TEXT_MODE(IGNORECASE).eq("GET")
   ```

2. Do not cache a response for a request that contains a body.

   The policy name is **_containsBody**. The following is the policy:

   ```
   (HTTP.REQ.HEADER("Content-Type").EXISTS &&
   HTTP.REQ.HEADER("Content-Type").eq("multipart/byteranges")) ||
   (HTTP.REQ.HEADER("Transfer-Encoding").EXISTS) ||
   (HTTP.REQ.HEADER("Content-Length").EXISTS)
   ```

3. Set a NOCACHE action for a request that contains an Authorization header.

The authorization header contains security-related information and should not be cached. The policy name is **_authorizationHeaderExists**. The following is the policy rule:

```
HTTP.REQ.HEADER("Authorization").EXISTS
```

4. Set a NOCACHE action for a request that contains a Proxy-Authorization header.

   This header contains security-related information and should not be cached. The policy name is _proxyAuthorizationHeaderExists. The following is the policy rule:

   ```
   HTTP.REQ.HEADER("Proxy-Authorization").EXISTS
   ```

5. Set a MAY_NOCACHE action for a request with a Cookie header.

   This header often contains information that is specific to the user and should not be cached. Another built-in policy that is evaluated later provides an exception by removing cookies from images and caching them. The policy name is **_cookieExists**. The following is the policy rule:

   ```
   HTTP.REQ.HEADER("Cookie").EXISTS
   ```

Default Response Policies

You can override the following default response-time policies by configuring new policies and binding them to the response-time override processing point.

The following policies are bound to the **_resBuiltinDefaults** policy label and are evaluated in the order in which they are listed:

1. Do not cache HTTP responses unless they are of type 200, 203, 300, 301, 302, 304, 307, 403, 404, or 410.

   The policy name is **_nonCachebleResponseStatus**. The following is the policy rule:

   ```
   ! ((HTTP.RES.STATUS.EQ(200)) || (HTTP.RES.STATUS.EQ(304)) ||
   (HTTP.RES.STATUS.EQ(203))|| (HTTP.RES.STATUS.EQ(300))||
   (HTTP.RES.STATUS.EQ(301))|| (HTTP.RES.STATUS.EQ(410))||
   (HTTP.RES.STATUS.EQ(404))|| (HTTP.RES.STATUS.EQ(403))||
   (HTTP.RES.STATUS.EQ(302))|| (HTTP.RES.STATUS.EQ(307)))
   ```

2. Do not cache an HTTP response if it has a Vary header with a value of anything other than Accept-Encoding.

   The compression module inserts the Vary: Accept_Encoding header. The name of this expression is **_nonAcceptEncoding**. The following is the policy rule:

   ```
   ((HTTP.RES.HEADER("Vary").EXISTS) && ((!
   HTTP.RES.HEADER("Vary").INSTANCE(1).LENGTH.eq(0)) || (!
   (HTTP.RES.HEADER("Vary").VALUE(0).STRIP_END_WS.eq("Accept-Encoding")))))
   ```

3. Do not cache a response if its Cache-Control header value is No-Cache, No-Store, or Private, or if the Cache-Control header is not valid.

   The policy name is **_cacheCntrlRespHeadersNoCache**. The following is the policy rule:

   ```
   ((HTTP.RES.CACHE_CONTROL.IS_PRIVATE) ||
   (HTTP.RES.CACHE_CONTROL.IS_NO_CACHE) ||
   (HTTP.RES.CACHE_CONTROL.IS_NO_STORE) ||
   (HTTP.RES.CACHE_CONTROL.IS_INVALID))
   ```

4. Cache responses if the Cache-Control header has one of the following values: Public, Must-Revalidate, Proxy-Revalidate, Max-Age, S-Maxage.

   The policy name is **_cacheCntrlRespHeadersCache**. The following is the policy rule:

   ```
   ((HTTP.RES.CACHE_CONTROL.IS_PUBLIC) ||
   (HTTP.RES.CACHE_CONTROL.IS_MAX_AGE) ||
   (HTTP.RES.CACHE_CONTROL.IS_MUST_REVALIDATE) ||
   (HTTP.RES.CACHE_CONTROL.IS_PROXY_REVALIDATE) ||
   (HTTP.RES.CACHE_CONTROL.IS_S_MAXAGE))
   ```

5. Do not cache responses that contain a Pragma header.

   The name of the policy is **_pragmaHeaderExists**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Pragma").EXISTS
   ```

6. Cache responses that contain an Expires header.

   The name of the policy is **_expiresHeaderExists**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Expires").EXISTS
   ```

7. If the response contains a Content-Type header with a value of Image, remove any cookies in the header and cache it.

   The name of the policy is **_contentImage**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Content-Type").EXISTS &&
   HTTP.RES.HEADER("Content-Type").VALUE(0).STARTSWITH("image/")
   ```

   You could configure the following content group to work with this policy:

   ```
   add cache contentgroup nocookie_group -removeCookies YES
   ```

8. Do not cache a response that contains a Set-Cookie header.

   The name of the policy is **_setCookieExists**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Set-Cookie").EXISTS ||
   HTTP.RES.HEADER("Set-Cookie2").EXISTS
   ```

Restrictions on Default Policies

You cannot override the following built-in request time policies with user-defined policies.

These policies are listed in priority order.

1. Do not cache any responses if the corresponding HTTP request lacks a GET or POST method.
2. Do not cache any responses for a request if the HTTP request URL length plus host name exceeds 1744 byes.
3. Do not cache a response for a request that contains an If-Match header.
4. Do not cache a request that contains an If-Unmodified-Since header.

   Note that this is different from the If-Modified-Since header.
5. Do not cache a response if the server does not set an expiry header.

You cannot override the following built-in response time policies. These policies are evaluated in the order in which they are listed:

1. Do not cache responses that have an HTTP response status code of 201, 202, 204, 205, or 206.
2. Do not cache responses that have an HTTP response status code of 4xx, with the exceptions of status codes 403, 404, and 410.
3. Do not cache responses if the response type is FIN terminated, or the response does not have one of the following attributes: Content-Length, or Transfer-Encoding: Chunked.
4. Do not cache the response if the caching module cannot parse its Cache-Control header.

Initial Settings for the Default Content Group

When you first enable integrated caching, the NetScaler appliance provides one predefined content group named the Default content group. The following table shows the settings for this group.

Table 1. Predefined Settings for the Default Content Group

Parameter	Description	Default Value
Hit parameters	The hit parameters contain the parameter names that are significant for generating a response. In parameterized hit selection, NetScaler appliance matches the URL stem byte-for-byte, matches normalized values of the hit parameters, and matches the target service information.	none
Invalidation Parameters	These parameters mark a cached object as obsolete during parameterized selection. Specific objects, or all objects in a content group, are selected if the values of the invalidation parameters in the object and in the request are same after normalization. The invalidation parameters are a subset of the hit parameters.	none
Poll Every Time	Poll every time for the objects in this content group.	NO

Ignore reload request	Specifies whether a request can force the system to reload a cached object from the origin. To guard against Denial of Service attacks, you must set this flag to YES. To get RFC-compliant behavior you should set it to NO.	YES
Remove Response Cookies	If this option is disabled for a content group, and if the response contains cookies, the cookies are stored and served with every cache hit. By default, the remove cookies option is enabled for a content group, to prevent the integrated cache from storing any responses with Set-Cookie or Set-Cookie2 headers unless the response is an image.	YES
Prefetch	The Prefetch option refreshes an object when it is about to expire. This ensures that the object remains stale or inactive (and therefore it cannot be served) for a shorter duration of time.	YES
Prefetch period	This duration in seconds during which prefetch should be attempted, immediately before the object's calculated expiry time.	heuristic
Maximum outstanding prefetches	The number of items that can be subjected to a prefetch at a time.	4294967295
Flashcache	Determines whether to enable queuing of client requests and simultaneous distribution of responses to all clients in the queue.	NO
Expire at last byte	Determines whether to expire a cached response immediately after serving it.	NO

Insert Via header	Defines a string to be inserted in a Via header. By default, a Via header is inserted in all responses served from a content group. The Via header is not inserted for responses that are served by the origin server.	YES
Insert Age header	The Age header contains information about the age of the object in seconds as calculated by the integrated cache.	YES
Insert ETag header	With ETag header insertion, the integrated cache does not serve full responses on repeat requests. This is done by forcing the user agent to cache the dynamic response sent by the cache the first time.	YES
Cache-control header	You can enable caching of dynamic objects in the user agent by replacing the Cache-Control headers that are inserted by the origin server. You must configure the new Cache-Control header to be inserted in the content group.	NONE
Quick abort size	If the size of an object that is being downloaded is less than or equal to the quick abort value, and a client aborts during the download, the cache stops downloading the response. If the object is larger than the quick abort size, the cache continues to download the response.	4194303 KBytes (maximum)
Minimum Response Size	You can control memory use by setting a minimum response size. Cached objects must be larger than the minimum response size.	0 KBytes

Maximum Response Size	You can control memory use by setting a maximum response size. Cached objects must be smaller than the maximum response size.	80 KBytes
Memory usage limit	Sets the maximum amount of memory that the cache can use. The effective limit is based on the available memory of the NetScaler appliance. The minimum value is 0 and the maximum value is unlimited.	UNLIMITED
Ignore caching headers in request	Disregards Cache-Control and Pragma headers in HTTP requests.	YES
MinHits configured	Number of hits that are required to qualify a response for storage in this content group.	0
Always evaluate policies		NO
Pinned	By default, when the cache is full the NetScaler appliance replaces the least recently used response first. The NetScaler appliance does not apply this behavior to content groups that are marked as pinned.	NO
Lazy DNS resolution	If set to YES, DNS resolution is performed for responses only if the destination IP address in the request does not match the destination IP address of the cached response.	YES

Link Load Balancing

Link load balancing (LLB) balances outbound traffic across multiple Internet connections provided by different service providers. LLB enables the Citrix® NetScaler® appliance to monitor and control traffic so that packets are transmitted seamlessly over the best possible link. Unlike with server load balancing, where a service represents a server, with LLB, a service represents a router or the next hop. A link is a connection between the NetScaler and the router.

To configure link load balancing, many users begin by configuring a basic setup with default settings. Configuring a basic setup involves configuring services, virtual servers, monitors, routes, an LLB method, and, optionally, configuring persistence. Once a basic setup is operational, you can customize it for your environment.

Load balancing methods that are applicable to LLB are round robin, destination IP hash, least bandwidth, and least packets. You can optionally configure persistence for connections to be sustained on a specific link. The available persistence types are source IP address-based, destination IP address-based, and source IP and destination IP address-based. PING is the default monitor but configuring a transparent monitor is recommended.

You can customize your setup by configuring reverse NAT (RNAT) and backup links.

Configuring a Basic LLB Setup

To configure LLB, you first create services representing each router to the Internet Service Providers (ISPs). A PING monitor is bound by default to each service. Binding a transparent monitor is optional but recommended. Then, you create a virtual server, bind the services to the virtual server, and configure a route for the virtual server. The route identifies the virtual server as the gateway to the physical routers represented by the services. The virtual server selects a router by using the load balancing method that you specify. Optionally, you can configure persistence to make sure that all traffic for a particular session is sent over a specific link.

Configuring a Basic LLB Setup

To configure LLB, you first create services representing each router to the Internet Service Providers (ISPs). A PING monitor is bound by default to each service. Binding a transparent monitor is optional but recommended. Then, you create a virtual server, bind the services to the virtual server, and configure a route for the virtual server. The route identifies the virtual server as the gateway to the physical routers represented by the services. The virtual server selects a router by using the load balancing method that you specify. Optionally, you can configure persistence to make sure that all traffic for a particular session is sent over a specific link.

A default monitor (PING) is automatically bound to a service when the service is created, but you can replace the default monitor with a transparent monitor, as described in Creating and Binding a Transparent Monitor.

To create a service by using the NetScaler command line

At the NetScaler command prompt, type:

- add service <name> <IP> <serviceType> <port>
- show service <name>

Example

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: DOWN
        Last state change was at Tue Aug 31 04:31:13 2010
        Time since last state change: 2 days, 05:34:18.600
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 244705  Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
```

Parameters for creating a service

name

The name of the service. Maximum length: 127

IP

The IP address of the physical router for which a service will be added.

serviceType

The type of connections that the service will handle. Specify a service type of ANY.

port

Port on which the service listens. Specify an asterisk (*) as the port number.

To create services by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. 3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service, as shown:
   - **Service Name*****—name
   - **Server**—IP
   - **Protocol*****—serviceType (Select **ANY** from the drop-down list.)
   - **Port*****—port

   * A required parameter
4. Click **Create**.
5. Repeat Steps 2-4 to create another service.
6. Click **Close**.
7. In the **Services** pane, select the services that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring an LLB Virtual Server and Binding a Service

After you create a service, create a virtual server and bind services to the virtual server. The default LB method of least connections is not supported in LLB. For information about changing the LB method, see Configuring the LLB Method and Persistence.

To create a link load balancing virtual server and bind a service by using the NetScaler command line

At the NetScaler command prompt, type:

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

Example

```
add lb vserver Router1-vip any
bind lb vserver Router-vip ISP1R_svc_any
sh lb vserver router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Thu Sep  2 10:51:32 2010
        Time since last state change: 0 days, 17:51:46.770
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED

1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
 Done
```

Parameters for creating an LLB virtual server

name

The name of the load balancing virtual server being added. Maximum length: 127

serviceType

The service type. Possible value: ANY.

Parameters for binding the service

name

The virtual server name to which the service is bound. Maximum length: 127

serviceName

The name of the service that is bound. Maximum Length: 127

To create a link load balancing virtual server and bind a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** pane, click **Add**.
3. In the **Create Virtual Servers (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating an LLB virtual server, as shown:
   - **Name***—name
   - **Protocol***—serviceType (Select **ANY**.)

   * A required parameter

   Note: Make sure **Directly Addressable** is unchecked.
4. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.
5. Click **Create**, and then click **Close**.
6. In the **Load Balancing Virtual Servers** tab, select the virtual server that you just created, and verify that the settings displayed in the **Details** pane are correct.

Configuring the LLB Method and Persistence

By default, the NetScaler appliance uses the least connections method to select the service for redirecting each client request, but you should set the LLB method to one of the supported methods. You can also configure persistence, so that different transmissions from the same client are directed to the same server.

To configure the LLB method and/or persistence by using the NetScaler command line

At the NetScaler command prompt, type the following command:

- set lb vserver <name> **-lbMethod** <lbMethod> **-persistenceType** <persistenceType>
- show lb vserver <name>

Example

```
set lb vserver router-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 00:52:21.200
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: SOURCEIP
     Persistence Mask: 255.255.255.255       Persistence v6MaskLength: 128   Persistence Timeout: 2 min
        Connection Failover: DISABLED
```

Parameters for configuring the LLB method and persistence

name

The name of the load balancing virtual server. Maximum Length: 127

lbMethod

The load balancing method. Possible values:

- ROUNDROBIN: Rotates the outgoing packets among the available links. This method distributes packets equally among the links, even if they operate at different speeds. Therefore, it can result in retransmissions or out-of-order packets.
- DESTINATIONHASH: Uses the hashed value of the destination IP address to select a link. You can mask the destination IP address to specify which part of it to use in the hash-value calculation, so that requests that are from different networks but destined for the same subnet are all directed to the same link.
- LEASTBANDWIDTH: Selects the link that is currently serving the least amount of traffic, measured in megabits per second (Mbps).
- LEASTPACKETS: Selects the link that has received the fewest packets in the last 14 seconds.

persistenceType

Persistence type for the virtual server. Possible values:

- SOURCEIP: Persistence based on the source IP address of inbound packets. After the load balancing method selects a link for transmission of the first packet, the NetScaler directs all subsequent packets sent from the same source IP address to the same link.
- DESTIP: Persistence based on the destination IP address of outbound packets. After the load balancing method selects a link for transmission of the first packet, the NetScaler directs all subsequent packets for the same destination IP address to the same link.
- SRCIPDESTIP: Persistence based on the source IP address of inbound packets and destination IP address of outbound packets. After the load balancing method selects a link for transmission of the first packet, the NetScaler directs all subsequent requests from the same source IP address and to the same destination IP address to the same link.

To configure the link load balancing method and/or persistence by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the Load Balancing method and/or persistence settings, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, specify values for some or all of the following parameters, which correspond to parameters described in "Parameters for configuring the LLB method and persistence" as shown:
   - **Method**—lbMethod
   - **Persistence**—persistenceType
4. Click **OK**.
5. In the **Load Balancing Virtual Servers** pane, select the virtual server that you just configured and verify that the settings displayed in the **Details** pane are correct.

Configuring an LLB Route

After configuring the services, virtual servers, LLB methods, and persistence, you configure an LLB route for the network specifying the virtual server as the gateway. A route is a collection of links that are load balanced. Requests are sent to the virtual server IP address that acts as the gateway for all outbound traffic and selects the router based on the LLB method configured.

To configure an LLB route by using the NetScaler command line

At the NetScaler command prompt, type:

- add lb route <network> <netmask> <gatewayName>
- show lb route [<network> <netmask>]

Example

```
add lb route 0.0.0.0 0.0.0.0 Router-vip
show lb route 0.0.0.0 0.0.0.0
     Network         Netmask      Gateway/VIP              State
    -----------     -------------  ------------------        --------
1)    0.0.0.0         0.0.0.0      Router-vip                UP
```

Parameters for configuring the LLB route

network

The IP address of the network to which the route belongs.

netmask

The mask specifying the subnet to which the route belongs.

gatewayName

The name of the virtual server. Maximum Length: 127

To configure an LLB route by using the configuration utility

1. In the navigation pane, expand Network, and then click **Routes**.
2. In the details pane, on the **LLB** tab, click **Add**.
3. In the **Create LB Route dialog** box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring the LLB route” as shown:
   - **Network***—network
   - **Netmask***—netmask
   - **Gateway Name***—gatewayName

   * A required parameter
4. Click **Create**, and then click **Close**. The route that you just created appears in the on the **LLB** tab in the **Routes** pane.

The following diagram shows a basic LLB setup. A service is configured for each of the two links (ISPs) and PING monitors are bound by default to these services. A link is selected based on the LLB method configured.

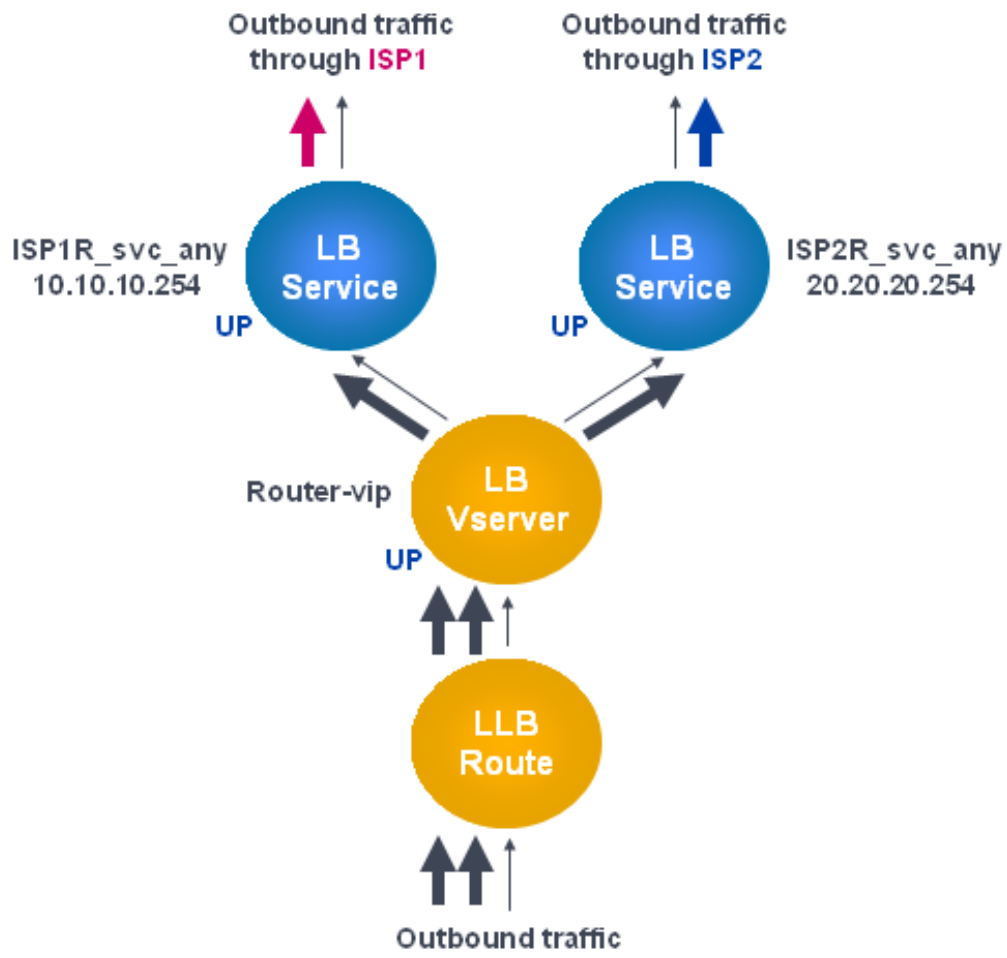


Figure 1. Basic LLB Setup

Creating and Binding a Transparent Monitor

You create a transparent monitor to monitor the health of upstream devices, such as routers. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the router is UP but one of the next hop devices from that router is down, the appliance includes the router while performing load balancing and forwards the packet to the router. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the router) are down, the service is marked as DOWN and the router is not included when the appliance performs link load balancing.

To create a transparent monitor by using the NetScaler command line

At the NetScaler command prompt, type:

- add lb monitor <monitorName> <type> **-destIP** <ip_addr|*> **-transparent** YES
- show lb monitor [<monitorName>]

Example

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)   Name.......: monitor-1  Type......:      PING   State....:   ENABLED
Standard parameters:
  Interval.........:          5 sec   Retries...........:             3
  Response timeout.:          2 sec   Down time.........:          30 sec
  Reverse..........:             NO   Transparent.......:           YES
  Secure...........:             NO   LRTM..............:       ENABLED
  Action...........:   Not applicable   Deviation.........:          0 sec
  Destination IP...:      10.10.10.11
  Destination port.:    Bound service
  Iptunnel.........:             NO
  TOS..............:             NO   TOS ID...........:             0
  SNMP Alert Retries:             0     Success Retries..:             1
  Failure Retries..:             0
```

Parameters for creating a transparent monitor

monitorName (Name)

The name of the monitor. Maximum Length: 31

type (Type)

The type of monitor.

destIP (Destination IP)

The IP address to which the probe is sent. If the destination IP address is set to 0, the destination IP address is that of the server to which the monitor is bound. Default value: 0

transparent (Transparent)

The state of the monitor for transparent devices, such as firewalls, based on the responsiveness of the services behind them. If the monitoring of transparent devices is enabled, the destination IP address should be specified. The probe is sent to the specified destination IP address by using the MAC address of the transparent device. Possible values: YES, NO. Default value: NO

To create a transparent monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the **Monitors** pane, click **Add**.
3. In the **Create Monitor** dialog box, set the following parameters:
   - **Name***
   - **Type***
   - **Destination IP**
   - **Transparent**

   * A required parameter
4. Click **Create**, and then click **Close**.
5. In the **Monitors** pane, select the monitor that you just configured and verify that the settings displayed in the **Details** pane are correct.

To bind a monitor to a service by using the NetScaler command line

At the NetScaler command prompt, type:

- bind lb monitor <monitorName> <serviceName>
- show service <serviceName>

Example

```
bind lb monitor monitor-HTTP-1 isp1R_svc_any
 Done
> show service isP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: UP
        Last state change was at Thu Sep  2 10:51:07 2010
        Time since last state change: 0 days, 18:41:55.130
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP  Weight: 1
                Probes: 1256       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
```

Parameters for binding a monitor

monitorName

The name of the monitor to be bound. Maximum Length: 31

serviceName

The name of the service or a service group to which the monitor is to be bound. Maximum Length: 127

To bind a monitor to a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select a service to which you want to bind a monitor, and then click **Open**.
3. In the **Configure Service** dialog box, on the **Monitors** tab, under **Available**, select the monitor that you want to bind to the service, and then click **Add**.
4. Click **OK**.
5. In the Services pane, select the service that you just configured and verify that the settings displayed in the **Details** pane are correct.

Configuring RNAT with LLB

You can configure an LLB setup for reverse network address translation (RNAT) for outbound traffic. This ensures that the return network traffic for a specific flow is routed through the same path. First configure basic LLB, as described in Configuring a Basic LLB Setup, and then configure RNAT. You must then enable use subnet IP (USNIP) mode.

To configure RNAT by using the NetScaler command line

At the NetScaler command prompt, type:

- set rnat <network> <netmask>
- show rnat

Example

```
set rnat 10.102.29.0 255.255.255.0
> show rnat
1)      Network: 10.102.29.0    Netmask: 255.255.255.0
        NatIP: *
```

Parameters for configuring RNAT

network

The network or subnet from which the traffic is flowing.

netmask

The subnet mask for the network

To configure RNAT by using the configuration utility

1. In the navigation pane, expand **Network**, and then click **Routes**.
2. In the detailspane, on the **RNAT** tab, click **Configure RNAT**.
3. In the **Configure RNAT** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring RNAT" as shown:
   - **Network***—network
   - **Netmask***—netmask

   * A required parameter
4. Click **Create**, and then click **Close**. The RNAT route that you just created appears in the on the **RNAT** tab in the **Routes** pane.

To enable Use Subnet IP mode by using the NetScaler command line

At the NetScaler command prompt, type:

- enable ns mode USNIP
- show ns mode

Example

```
enable ns mode USNIP
> show ns mode
      Mode                  Acronym          Status
      -------               -------          ------
 1)   Fast Ramp               FR               ON
 2)   ....
 8)   Use Subnet IP           USNIP            ON
 9)   ...
```

To enable Use Subnet IP mode by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In details pane, under **Modes and Features**, click **Configure modes**.
3. In the **Configure Modes** dialog box, select **Use Subnet IP**, and then click **OK**.
4. In the **Enable/Disable Mode(s)** message box, click **Yes**.

Configuring a Backup Route

To prevent disruption in services when the primary route is down, you can configure a backup route. Once the backup route is configured, the NetScaler appliance automatically uses it when the primary route fails. First create a primary virtual server as described in Configuring an LLB Virtual Server and Binding a Service. To configure a backup route, create a secondary virtual server similar to a primary virtual server and then designate this virtual server as a backup virtual server (route).

In the following diagram, **Router-vip** is the primary virtual server, and **Backup_Router-vip** is the secondary virtual server designated as the backup virtual server.

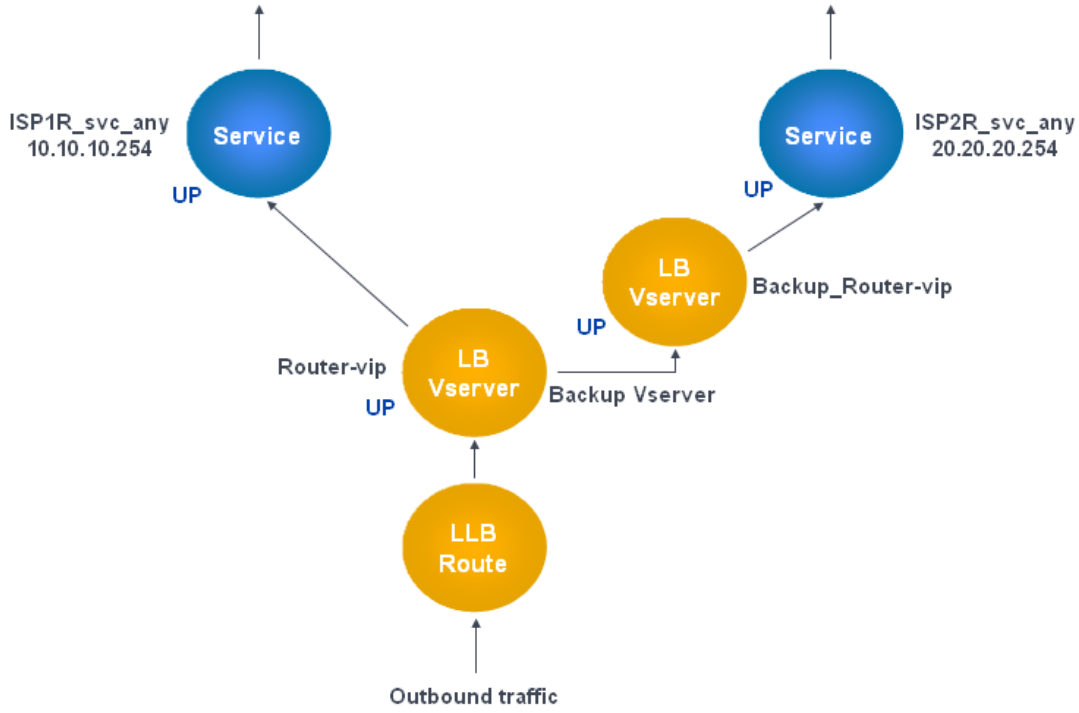


Figure 1. Backup Route Setup

By default, all traffic is sent through the primary route. However, when the primary route fails, all traffic is diverted to the backup route as shown in the following diagram.

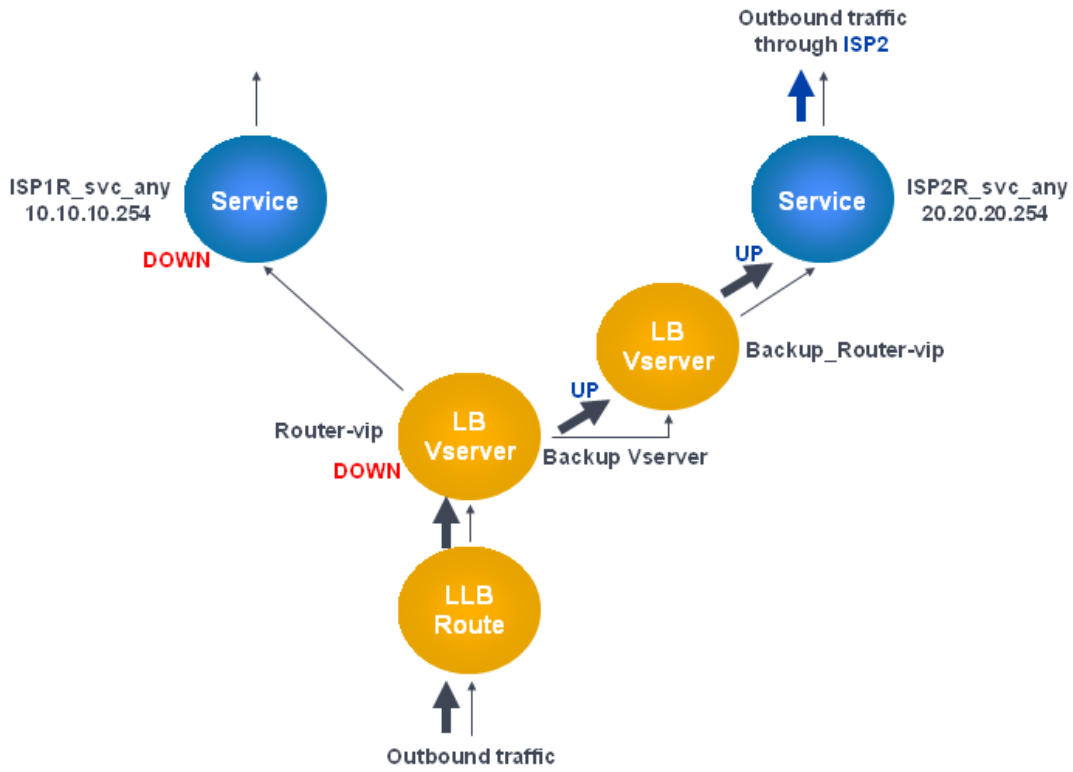


Figure 2. Backup Routing in Operation

To set the secondary virtual server as the backup virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <name> **-backupVserver** <string>

Example

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 03:09:45.600
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: ROUNDROBIN
```

```
    Mode: IP
    Persistence: DESTIP    Persistence Mask: 255.255.255.255    Persistence v6MaskLength: 128    Persiste
    Backup: Router2-vip
    Connection Failover: DISABLED
Done
```

Parameters for setting up the secondary virtual server as the backup virtual server

name

The name of the load balancing virtual server for which you are configuring a backup. Maximum Length: 127

backupVServer

The name of the backup virtual server. Maximum Length: 127

To set the secondary virtual server as the backup virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the secondary virtual server for which you want to configure the backup virtual server, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. In the **Backup Virtual Server** drop-down list, select the secondary backup virtual server, and then click **OK**.

Resilient LLB Deployment Scenario

In the following diagram, there are two networks: 30.30.30.0 and 30.30.31.0. Link load balancing is configured based on the destination IP address. Two routes are configured with gateways **Router1-vip** and **Router2-vip**, respectively. **Router1-vip** is configured as a backup to **Router2-vip** and vice versa. All traffic with the destination IP specified as 30.30.30.30 is sent through **Router1-vip** and traffic with the destination IP specified as 30.30.31.31 is sent through **Router2-vip**.

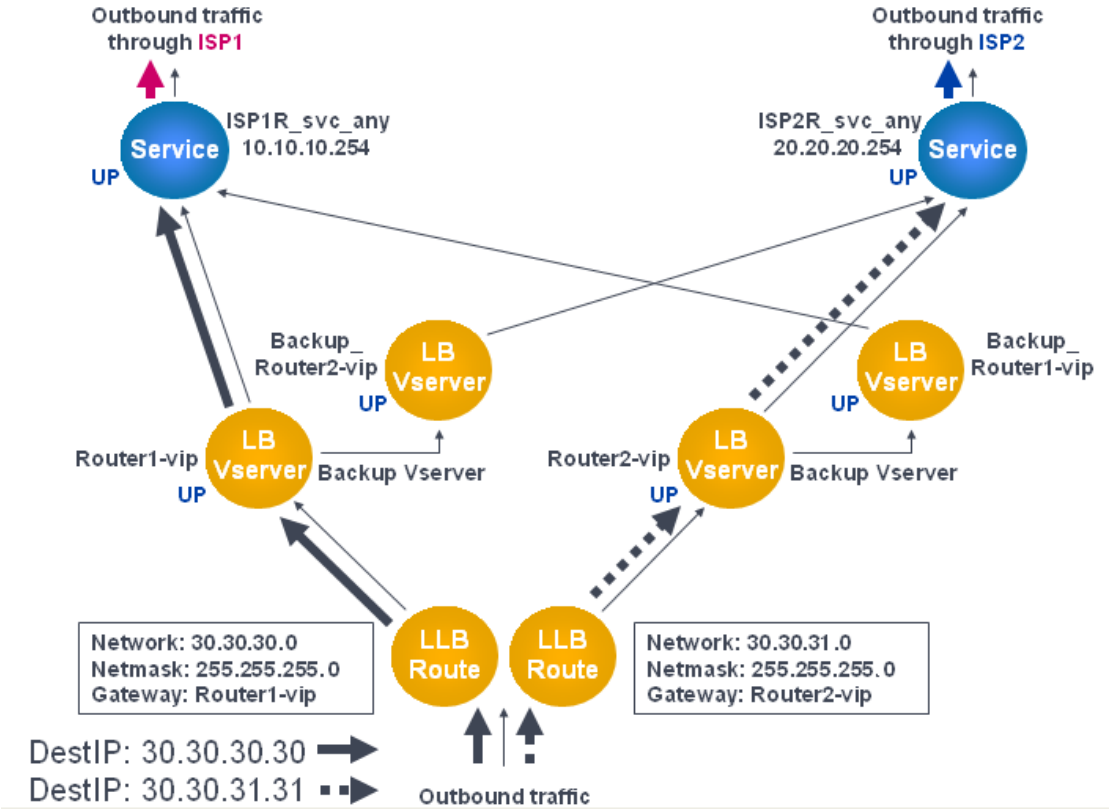


Figure 1. Resilient LLB Deployment Setup

However, if any one of the gateways (**Router1-vip** or **Router2-vip**) is DOWN, traffic is routed through the backup router. In the following diagram, **Router1-vip** for ISP1 is DOWN, so all traffic with the destination IP specified as 30.30.30.30 is also sent through ISP2.

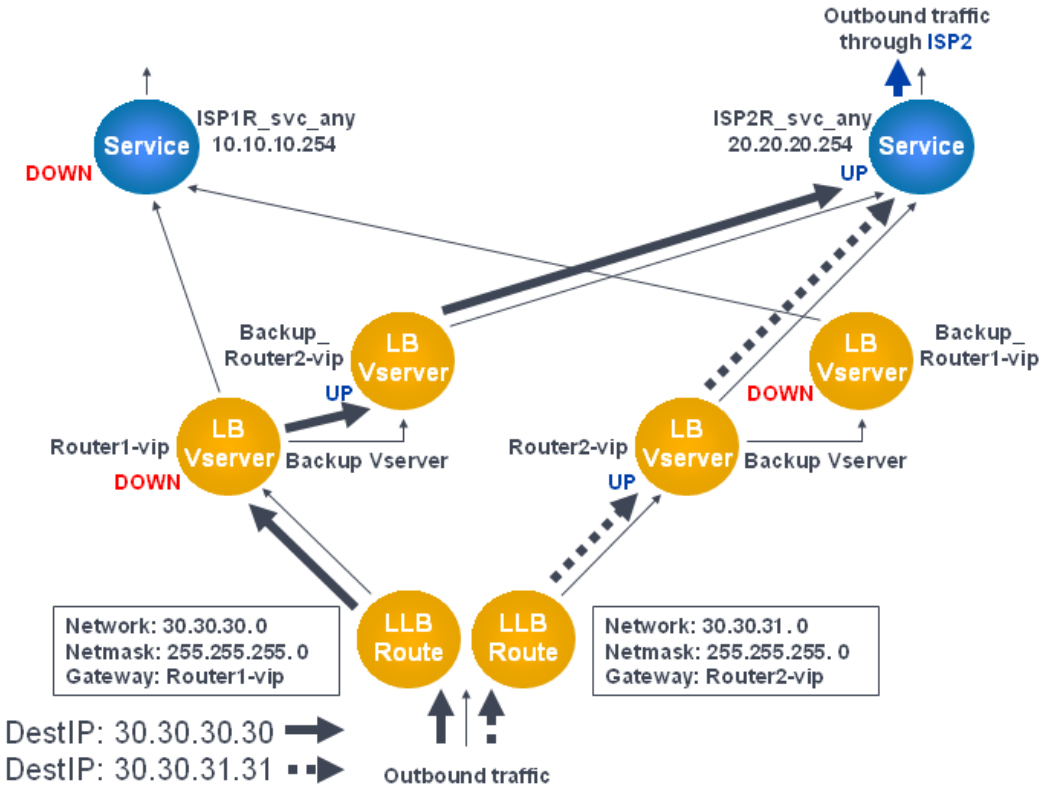


Figure 2. Resilient LLB Deployment Scenario

Monitoring an LLB Setup

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the NetScaler command line

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the NetScaler command prompt, type:

stat lb vserver [-**detail**] **[<name>]**

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
                  vsvrIP  port    Protocol      State   Req/s  Hits/s
One                    *    80        HTTP         UP     5/s     0/s
Two                    *     0         TCP       DOWN     0/s     0/s
Three                  *  2598         TCP       DOWN     0/s     0/s
```

```
dnsVirtualNS    10.102.29.90    53        DNS        DOWN      0/s      0/s
BRVSERV          10.10.1.1    80        HTTP       DOWN      0/s      0/s
LBVIP          10.102.29.66    80        HTTP         UP      0/s      0/s
 Done
```

Parameters for displaying statistics

detail

Include the statistics for hits per second and the total number of hits.

name

Name of the virtual server whose statistics are displayed.

To display virtual server statistics by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.
3. In the details pane, click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the NetScaler command line

At the NetScaler command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service whose statistics you want to view (for example, **Service-HTTP-1**).
3. Click **Statistics**. The statistics appear in a new window.

Load Balancing

The load balancing feature distributes user requests for Web site pages and other protected applications across multiple servers that all host (or mirror) the same content. You use load balancing primarily to manage user requests to heavily used applications, preventing poor performance and outages and ensuring that users can access your protected applications. Load balancing also provides fault tolerance; when one server that hosts a protected application becomes unavailable, the feature distributes user requests to the other servers that host the same application.

You can configure the load balancing feature to:

- Distribute all requests for a specific protected Web site, application, or resource between two or more identically-configured servers.
- Use any of several different algorithms to determine which server should receive each incoming user request, basing the decision on different factors, such as which server has the fewest current user connections or which server has the lightest load.

The load balancing feature is a core feature of the NetScaler appliance. Most users first set up a working basic configuration and then customize various settings, including persistence for connections. In addition, you can configure features for protecting the configuration against failure, managing client traffic, managing and monitoring servers, and managing a large scale deployment.

How Load Balancing Works

In a basic load balancing setup, clients send their requests to the IP address of a virtual server configured on the NetScaler appliance. The virtual server distributes them to the load-balanced application servers according to a preset pattern, called the load balancing algorithm. In some cases, you might want to assign the load balancing virtual server a wildcard address instead of a specific IP address. For instructions about specifying a global HTTP port on the NetScaler, see Global HTTP Ports.

How Load Balancing Works

In a basic load balancing setup, clients send their requests to the IP address of a virtual server configured on the NetScaler appliance. The virtual server distributes them to the load-balanced application servers according to a preset pattern, called the load balancing algorithm. In some cases, you might want to assign the load balancing virtual server a wildcard address instead of a specific IP address. For instructions about specifying a global HTTP port on the NetScaler, see Global HTTP Ports.

Load Balancing Basics

A load balancing setup includes a load-balancing server and multiple load-balanced application servers. The virtual server receives incoming client requests, uses the load balancing algorithm to select an application server, and forwards the requests to the selected application server. The following conceptual drawing illustrates a typical load balancing deployment. Another variation involves assigning a global HTTP port.

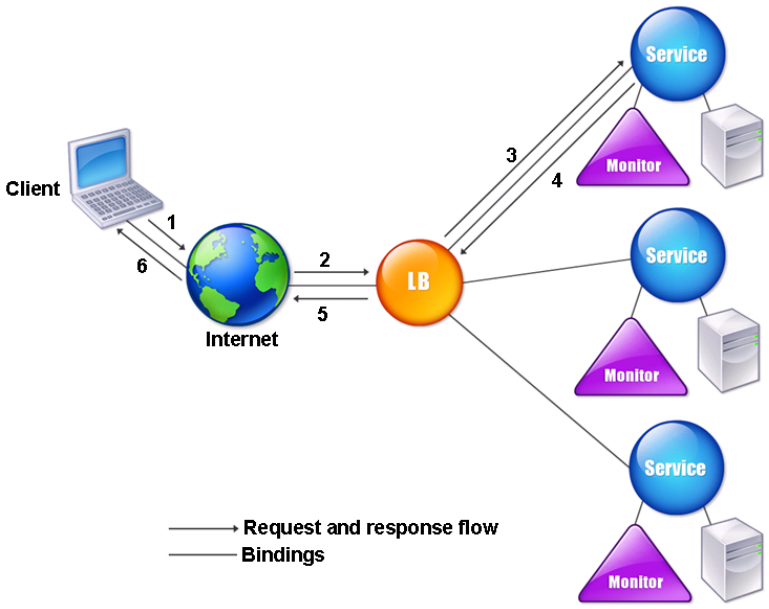


Figure 1. Load Balancing Architecture

The load balancing server can use any of a number of algorithms (or methods) to determine how to distribute load among the load-balanced servers that it manages. The default load balancing method is the least connection method, in which the NetScaler appliance forwards each incoming client connection to whichever load-balanced application server currently has the fewest active user connections.

The entities that you configure in a typical NetScaler load balancing setup are:

- **Load balancing virtual server.** The IP address, port, and protocol combination to which a client sends connection requests for a particular load-balanced web site or application. If the application is accessible from the Internet, the virtual server IP address (VIP) is a public IP address. If the application is accessible only from the local area network (LAN) or wide area network (WAN), the VIP is usually a private (ICANN non-routable) IP address.

- **Service.** The IP address, port, and protocol combination used to route requests to a specific load-balanced application server. A service can be a logical representation of the application server itself, or of an application running on a server that hosts multiple applications. Each service is bound to a specific virtual server.

- **Server object.** An entity that identifies a physical server and provides the server's IP address. If you want to use the server's IP address as the name of the server object, you can enter the server's IP address when you create a service, and the server object is then created automatically. Alternatively, you can create the server object first and assign it an FQDN or other name, and then specify that name instead of the IP address when you create the service.

- **Monitor.** An entity on the NetScaler appliance that tracks a service and ensures that it is operating correctly. The monitor periodically probes (or performs a health check on) each service to which you assign it. If the service does not respond within the time specified by the time-out, and a specified number of health checks fail, that service is marked DOWN. The NetScaler appliance then skips that service when performing load balancing, until the issues that caused the service to quit responding are fixed.

The virtual server, services, and load balanced application servers in a load balancing setup can use either Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) IP addresses. You can mix IPv4 and IPv6 addresses in a single load balancing setup.

For variations in the load balancing setup, see the following use cases:

- Configuring Load Balancing in Direct Server Return Mode
- Configuring LINUX Servers in DSR Mode
- Configuring DSR Mode When Using TOS
- Configuring Load Balancing in DSR Mode by Using IP Over IP
- Configuring Load Balancing in One-arm Mode
- Configuring Load Balancing in the Inline Mode
- Load Balancing of Intrusion Detection System Servers
- Load Balancing RDP services

Understanding the Topology

In a load balancing setup, the load balancing server is logically located between the client and the server farm, and manages traffic flow to the servers in the server farm. On the NetScaler appliance, the application servers are represented by virtual entities called services. The following diagram shows the topology of a basic load balancing configuration.

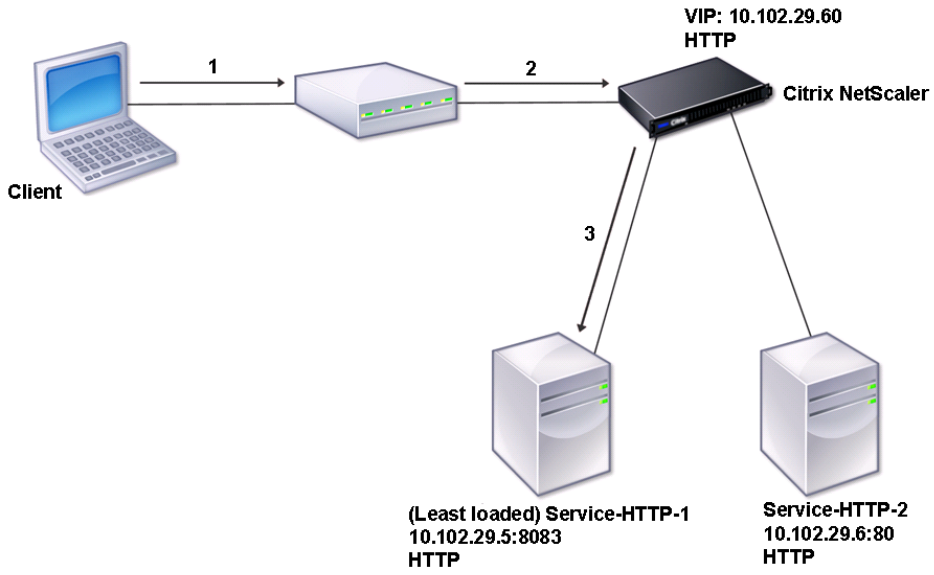


Figure 1. Basic Load Balancing Topology

In the diagram, load balancing is used to manage traffic flow to the servers. The virtual server selects the service and assigns it to serve client requests. Consider a scenario where the services Service-HTTP-1 and Service-HTTP-2 are created and bound to the virtual server named Vserver-LB-1. Vserver-LB-1 forwards the client request to either Service-HTTP-1 or Service-HTTP-2. The NetScaler appliance uses the least connection load balancing method to select the service for each request. The following table lists the names and values of the basic entities that must be configured on the appliance.

Entity	Mandatory Parameters and Sample Values			
	Name	IP Address	Port	Protocol
Virtual server	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	8083	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP

Monitors	Default	None	None	None

The following diagram shows the load balancing sample values and mandatory parameters that are described in the preceding table.

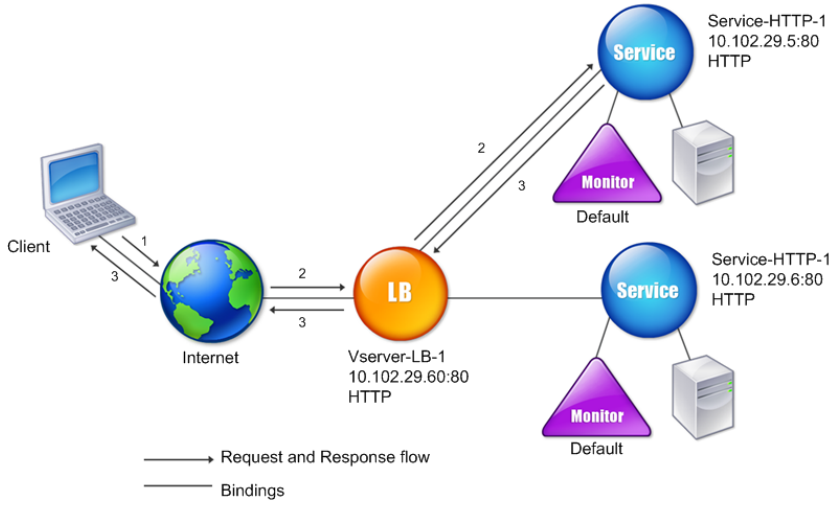


Figure 2. Load Balancing Entity Model

Use of Wildcards Instead of IP Addresses and Ports

In some cases you might need to use a wildcard for the IP address or the port of a virtual server or for the port of a service. The following cases may require using a wildcard:

- If the NetScaler appliance is configured as a transparent pass through, which must accept all traffic that is sent to it regardless of the IP or port to which it is sent.
- If one or more services listen on ports that are not well known.
- If one or more services, over time, change the ports that they listen on.
- If you reach the limit for the number of IP addresses and ports that you can configure on a single NetScaler appliance.
- If you want to create virtual servers that listen for all traffic on a specific virtual LAN.

When a wildcard-configured virtual server or service receives traffic, the NetScaler appliance determines the actual IP address or port and creates new records for the service and associated load balanced application server. These dynamically created records are called dynamically learned server and service records.

For example, a firewall load balancing configuration can use wildcards for both the IP address and port. If you bind a wildcard TCP service to this type of load balancing virtual server, the virtual server receives and processes all TCP traffic that does not match any other service or virtual server.

The following table describes some of the different types of wildcard configurations and when each should be used.

IP	Port	Protocol	Description

*	*	TCP	A general wildcard virtual server that accepts traffic sent to any IP address and port on the NetScaler appliance. When using a wildcarded virtual server, the appliance dynamically learns the IP and port of each service and creates the necessary records as it processes traffic.
*	*	TCP	A firewall load balancing virtual server. You can bind firewall services to this virtual server, and the NetScaler appliance passes traffic through the firewall to the destination.

IP Address	*	TCP,UDP, and ANY	A virtual server that accepts all traffic that is sent to the specified IP address, regardless of the port. You must explicitly bind to this type of virtual server the services to which it will redirect traffic. It will not dynamically learn them. **Note:** You do not configure services or virtual servers for a global HTTP port. In this case, you configure a specific port as a global HTTP port (for example, set ns config -httpPort 80). The appliancee then accepts all traffic that matches the port number, and processes it as HTTP traffic. The appliance dynamically learns and creates services for this traffic.

*	port	SSL, SSL_TCP	A virtual server that accepts all traffic sent to any IP address on a specific port. Used for global transparent SSL offloading. All SSL, HTTP, and TCP processing that usually is performed for a service of the same protocol type is applied to traffic that is directed to this specific port. The appliance uses the port to dynamically learn the IP of the service it should use. If -cleartext is not specified, the NetScaler appliance uses end-to-end SSL.
*	port	Not applicable	All other virtual servers that can accept traffic to the port. You do not bind services to these virtual servers; the NetScaler appliance learns them dynamically.

Note: If you have configured your NetScaler appliance as a transparent pass through that makes use of global (wildcard) ports, you may want to turn on Edge mode. For more information, see Configuring Edge Mode.

The NetScaler appliance attempts to locate virtual servers and services by first attempting an exact match. If none is found, it continues to search for a match based on wildcards, in the following order:

1. Specific IP address and specific port number
2. Specific IP address and a * (wildcard) port
3. * (wildcard) IP address and a specific port
4. * (wildcard) IP address and a * (wildcard) port

If the appliance is unable to select a virtual server by IP address or port number, it searches for a virtual server on the basis of the protocol used in the request, in the following order:

1. HTTP
2. TCP
3. ANY

Global HTTP Ports

You do not configure services or virtual servers for a global HTTP port. Instead, you configure a specific port by using the following command:

To configure a global http port by using the NetScaler command line

At the NetScaler command prompt, type:

set ns config –**httpPort** <port>

Example

set ns config –httpPort 80

After configuring this port, the NetScaler appliance accepts all traffic that matches the port number, and processes it as HTTP traffic, dynamically learning and creating services for that traffic.

Setting Up Basic Load Balancing

Before configuring your initial load balancing setup, enable the load balancing feature. Then begin by creating at least one service for each server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server, and bind each service to the virtual server. That completes the initial setup. Before proceeding with further configuration, verify your configuration to make sure that each element was configured properly and is operating as expected.

Enabling Load Balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the NetScaler command line

At the NetScaler command prompt, type the following command to enable load balancing and verify the configuration:

- enable feature lb
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               ON
3)    Load Balancing           LB               ON
.
.
.
24)   NetScaler Push           push             OFF
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. In the **Configure Basic Features** dialog box, select the **Load Balancing** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Configuring Services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the NetScaler appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

Adding a Server

If you add a server to the list of servers before creating a service to represent that server, you can assign a name to the server. You can then specify the server's name instead of its IP address when you create a service. If you create a service from the configuration utility, you can select the server from the drop-down list. When adding the server to the list, you might want to assign it a name that helps identify the kind of services that you will create with it. You can also specify its state and add a comment.

When adding a server, you must identify it by specifying its IP address or domain name. If you specify the domain name, you can later change the IP address of the physical server without having to modify the server's entry in the list of servers on the NetScaler. The domain name is resolved to an IP address that is specified in an address record on the DNS.

During hardware maintenance or software upgrades, you may have to make the server DOWN. If the server is domain based, to override the IP address resolved by the DNS, you can configure an IP address mask and a translation IP address on the NetScaler. For more information, see Translating the IP Address of a Domain Based Server.

Note: You can add a range of servers from a single CLI command or the same dialog box. The names in the range vary by a number used as a suffix or prefix. For example, server1, server2, and so on. From the configuration utility, you can specify a range only in the last octet of the IP address, which is the fourth in case of an IPv4 address and the eighth in case of an IPv6 address. From the command line, you can specify the range in any octet of the IP address.

To add a server by using the NetScaler command line

At the NetScaler command prompt, type:

add server <name> (<ipAddress> | (<domain> [**-ipv6Address** (YES | NO)]) [**-state** (ENABLED | DISABLED)] [**-comment** <string>]

Examples

```
add server satexam www.satexam.net -state DISABLED -comment ServerforSATResults
Done
add server Server-1 10.102.29.18
Done
```

Parameters for configuring a server

name

The name assigned to the server. This alphanumeric string is required. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ipAddress

IP address of the server, in either IPv4 or IPv6 format.

If the server is not reachable from the NetScaler appliance or is not active, the service is marked as DOWN.

domain

Domain name that resolves to the IP address that represents the server.

ipv6Address

Resolve the domain name to an IPv6 address. Possible values: YES, NO. Default: NO.

state

The initial state of the server. Possible values: ENABLED, DISABLED. Default: ENABLED.

comment

A comment to help identify the server. Maximum length: 255 characters. To include spaces in a comment that you type on the NetScaler command line, enclose the entire comment inside quotation marks. The quotation marks become part of the comment. They are not required if you use the configuration utility.

To add a server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Servers**.
2. In the details pane, click **Add**.
3. In the **Create Server** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a server" as shown above:

   To add a server by using the configuration utility

   - **Server Name**—name
   - **IP Address**—ipAddress (Select **IP Address** and type the address. Before typing an IPv6 address, select the **IPv6** check box.)
   - **Domain Name**—domain (For a domain-name based server, select **Domain Name** and type the name of the server's domain.)
   - **Enable after Creating**—state
   - **Comment**—comment
4. If you specify the domain name of the server and you want the domain name to be resolved to an IPv6 address, select the **IPv6 Domain** check box.
5. Click **Create**, and then click **Close**. The server you named appears in the **Servers** pane.

Creating a Service

Before you create a service, you need to understand the different service types and how each is used. The following table describes the types of services supported on the NetScaler appliance.

HTTP

Used for load-balanced servers that accept HTTP traffic, such as standard Web sites and Web applications. The HTTP service type enables the NetScaler appliance to provide compression, content filtering, caching, and client Keep alive support for your Layer 7 Web servers. This service type also supports virtual server IP port insertion, redirect port rewriting, Web 2.0 PUSH, and URL redirection support.

Because HTTP is a TCP-based application protocol, you can also use the TCP service type for Web servers. If you do so, however, the NetScaler appliance is able to perform only Layer 4 load balancing. It cannot provide any of the Layer 7 support described above.

SSL

Used for servers that accept HTTPS traffic, such as ecommerce Web sites and shopping cart applications. The SSL service type enables the NetScaler appliance to encrypt and decrypt SSL traffic (perform SSL offloading) for your secure Web applications. It also supports HTTP persistence, content switching, Rewrite, virtualserver IP port insertion, Web 2.0 PUSH, and URL redirection.

You can also use the SSL_BRIDGE, SSL_TCP, or TCP service types. If you do so, however, the NetScaler performs only Layer 4 load balancing. It cannot provide SSL offloading or any of the Layer 7 support described above

FTP

Used for servers that accept FTP traffic. The FTP service type enables the NetScaler appliance to support specific details of the FTP protocol.

You can also use TCP or ANY service types for FTP servers.

TCP

Used for servers that accept many different types of TCP traffic, or that accept a type of TCP traffic for which a more specific type of service is not available.

You can also use the ANY service type for these servers.

SSL_TCP

Used for servers that accept non-HTTP-based SSL traffic, to support SSL offloading.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance performs Layer 4 load balancing, but not SSL offloading.

UDP

Used for servers that accept UDP traffic. You can also use the ANY service type.

SSL_BRIDGE

Used for servers that accept SSL traffic when you do not want the NetScaler appliance to perform SSL offloading. Alternatively, you can use the SSL_TCP service type.

NNTP

Used for servers that accept Network News Transfer Protocol (NNTP) traffic, typically Usenet sites.

DNS

Used for servers that accept DNS traffic, typically nameservers. With the DNS service type, the NetScaler appliance validates the packet format of each DNS request and response. It can also cache DNS responses. You can apply DNS policies to DNS services.

You can also use the UDP service type for these services. If you do, however, the NetScaler appliance can only perform Layer 4 load balancing. It cannot provide support for DNS-specific features.

ANY

Used for servers that accept any type of TCP, UDP, or ICMP traffic. The ANY parameter is used primarily with firewall load balancing and link load balancing.

SIP-UDP

Used for servers that accept UDP-based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP).

You can also use the UDP service type for these services. If you do, however, the NetScaler appliance performs only Layer 4 load balancing. It cannot provide support for SIP-specific features.

DNS-TCP

Used for servers that accept DNS traffic, where the NetScaler appliance acts as a proxy for TCP traffic sent to DNS servers. With services of the DNS-TCP service type, the NetScaler appliance validates the packet format of each DNS request and response and can cache DNS responses, just as with the DNS service type.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance only performs Layer 4 load balancing of external DNS name servers. It cannot provide support for any DNS-specific features.

RTSP

Used for servers that accept Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data. Select this type to support audio, video, and other types of streamed media.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance performs only Layer 4 load balancing. It cannot parse the RTSP stream or provide support for RTSPID persistence or RTSP NATting.

DHCPRA

Used for servers that accept DHCP traffic. The DHCPRA service type can be used to relay DHCP requests and responses between VLANs.

Services are designated as DISABLED until the NetScaler appliance connects to the associated load-balanced server and verifies that it is operational. At that point, the service is designated as ENABLED. Thereafter, the NetScaler appliance periodically monitors the status of the servers, and places any that fail to respond to monitoring probes (called health checks) back in the DISABLED state until they respond.

Note: You can create a range of services from a single CLI command or the same dialog box. The names in the range vary by a number used as a suffix/prefix. For example, service1, service2, and so on. From the configuration utility, you can specify a range only in the last octet of the IP address, which is the fourth in case of an IPv4 address and the eighth in case of an IPv6 address. From the command line, you can specify the range in any octet of the IP address.

To create a service by using the NetScaler command line

At the NetScaler command prompt, type:

add service <name> <serverName> <serviceType> <port>

Example

add service Service-HTTP-1 10.102.29.5 HTTP 80

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server that hosts this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

Note: For more information about the SSL and SSL_TCP service types, see SSL Offload and Acceleration.

To create a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a service” as shown:
   - **Service Name**—name
   - **Server**—serverName
   - **Protocol**—serviceType
   - **Port**—port
4. Click **Create**, and then click **Close**. The service you created appears in the **Services** pane.

Creating a Virtual Server

After you create your services, you must create a virtual server to accept traffic for the load balanced Web sites, applications, or servers. Once load balancing is configured, users connect to the load-balanced Web site, application, or server through the virtual server's IP address or FQDN.

Note: The virtual server is designated as DOWN until you bind the services that you created to it, and until the NetScaler appliance connects to those services and verifies that they are operational. Only then is the virtual server designated as UP.

To create a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> <serviceType> <ip> <port>

Example

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

Parameters for creating a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, RDP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

To create a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a virtual server” as shown:
   - **Name**—name
   - **IP Address**—IPAddress

     Note: If the virtual server uses IPv6, select the **IPv6** check box and enter the address in IPv6 format (for example, **1000:0000:0000:0000:0005:0600:700a:888b**).
   - **Protocol**—serviceType
   - **Port**—port
4. Click **Create**, and then click **Close**. The virtual server you created appears in the **Load Balancing Virtual Servers** pane.

Binding Services to the Virtual Server

After you have created services and a virtual server, you must bind the services to the virtual server. In most cases, services are bound to virtual servers of the same type, but you can bind certain types of services to certain different types of virtual servers, as shown below.

Virtual Server Type	Service Type	Comment
HTTP	SSL	You would normally bind an SSL service to an HTTP virtual server to do encryption.
SSL	HTTP	You would normally bind an HTTP service to an SSL virtual server to do SSL offloading.
SSL_TCP	TCP	You would normally bind a TCP service to an SSL_TCP virtual server to do SSL offloading for other TCP (SSL decryption without content awareness).

The state of the services bound to a virtual server determines the state of the virtual server: if all of the bound services are DOWN, the virtual server is marked DOWN, and if any of the bound services is UP or OUT OF SERVICE, the state of the virtual server is UP.

To bind a service to a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <name> <serviceName>

Example

bind lb vserver Vserver-LB-1 Service-HTTP-1

To bind a service to a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service (for example, **Vserver-LB-1**).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server (for example, **Service-HTTP-1**).
5. Click **OK**.

Note: You can bind a service to multiple virtual servers.

Verifying the Configuration

After finishing your basic configuration, you should view the properties of each service and load balancing virtual server in your load balancing setup to verify that each is configured correctly. After the configuration is up and running, you should view the statistics for each service and load balancing virtual server to check for possible problems.

Viewing the Properties of a Server Object

You can view properties such as the name, state, and IP address of any server object in your NetScaler appliance configuration.

To view the properties of server objects by using the NetScaler command line

At the NetScaler command prompt, type:

show server <serverName>

Example

show server server-1

To view the properties of server objects by using the configuration utility

In the navigation pane, expand **Load Balancing**, and then click **Servers**. The parameter values of the available servers appear in the details pane.

Viewing the Properties of a Virtual Server

You can view properties such as the name, state, effective state, IP address, port, protocol, method, and number of bound services for your virtual servers. If you have configured more than the basic load balancing settings, you can view the persistence settings for your virtual servers, any policies that are bound to them, and any cache redirection and content switching virtual servers that have been bound to the virtual servers.

Note: For information on cache redirection, see Cache Redirection. For information on content switching, see Content Switching.

To view the properties of a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

show lb vserver <name>

Example

show lb vserver Vserver-LB-1

To view the properties of a load balancing virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click a virtual server to display its properties at the bottom of the details pane.
3. To view cache redirection and content switching virtual servers that are bound to this virtual server, click **Show CS/CR Bindings**.

Viewing the Properties of a Service

You can view the name, state, IP address, port, protocol, maximum client connection, maximum requests per connection, and server type of the configured services, and use this information to troubleshoot any mistake in the service configuration.

To view the properties of services by using the NetScaler command line

At the NetScaler command prompt, type:

show service <name>

Example

show service Service-HTTP-1

To view the properties of services by using the configuration utility

In the navigation pane, expand **Load Balancing**, and then click **Services**. The details of the available services appear on the **Services** pane.

Viewing the Bindings of a Service

You can view the list of virtual servers to which the service is bound. The binding information also provides the name, IP address, port and state of the virtual servers to which the services are bound. You can use the binding information to troubleshoot any problem with binding the services to virtual servers.

To view the bindings of a service by using the NetScaler command line

At the NetScaler command prompt, type:

show service bindings <name>

Example

show service bindings Service-HTTP-1

To view the bindings of a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service whose binding information you want to view (for example, **Service-HTTP-1**).
3. Click **Show Bindings**. The bindings of the service you selected appear in the **Binding details for Service: ServiceName** dialog box.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the NetScaler command line

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the NetScaler command prompt, type:

stat lb vserver [-**detail**] **[<name>]**

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
                  vsvrIP  port     Protocol       State    Req/s   Hits/s
One                  *    80         HTTP          UP      5/s      0/s
Two                  *     0          TCP        DOWN      0/s      0/s
Three                 *  2598          TCP        DOWN      0/s      0/s
dnsVirtualNS    10.102.29.90    53          DNS        DOWN      0/s      0/s
BRVSERV          10.10.1.1    80         HTTP        DOWN      0/s      0/s
LBVIP         10.102.29.66    80         HTTP          UP      0/s      0/s
 Done
```

Parameters for displaying statistics

detail

Include the statistics for hits per second and the total number of hits.

name

Name of the virtual server whose statistics are displayed.

To display virtual server statistics by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.
3. In the details pane, click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the NetScaler command line

At the NetScaler command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service whose statistics you want to view (for example, **Service-HTTP-1**).
3. Click **Statistics**. The statistics appear in a new window.

Customizing a Load Balancing Configuration

After you configure a basic load balancing setup, you can make a number of modifications to it so that it distributes load exactly as you need. The load balancing feature is complex. You can modify the basic elements by changing the load balancing algorithm, configuring load balancing groups and using them to create your load balancing configuration, configuring persistent client-server connections, configuring the redirection mode, and assigning different weights to different services that have different capacities.

The default load balancing algorithm on the NetScaler appliance is the least connection method, which configures the appliance to send each incoming connection to the service that is currently handling the fewest connections. You can specify different load balancing algorithms, each of which is suited to different conditions.

To accommodate applications such as shopping carts, which require that all requests from the same user be directed to the same server, you can configure the appliance to maintain persistent connections between clients and servers. You can also specify persistence for a group of virtual servers, causing the appliance to direct individual client requests to the same service regardless of which virtual server in the group receives the client request.

You can enable and configure the redirection mode that the appliance uses when redirecting user requests, choosing between IP-based and MAC-based forwarding. You can assign weights to different services, specifying what percentage of incoming load should be directed to each service, so that you can include servers with different capacities in the same load balancing setup without overloading the lower-capacity servers or allowing the higher-capacity servers to sit idle.

Load Balancing Algorithms

The load balancing algorithm defines the criteria that the NetScaler appliance uses to select the service to which to redirect each client request. Different load balancing algorithms use different criteria. For example, the least connection algorithm selects the service with the fewest active connections, while the round robin algorithm maintains a running queue of active services, distributes each connection to the next service in the queue, and then sends that service to the end of the queue.

Some load balancing algorithms are best suited to handling traffic on ordinary Web sites, others to managing traffic to DNS servers, and others to handling complex Web applications used in e-commerce or on company LANs or WANs. The following table lists each load balancing algorithm that the NetScaler appliance supports, with a brief description of how each operates.

Name	Server Selection Based On
LEASTCONNECTION	Which service currently has the fewest client connections. This is the default load balancing algorithm.
ROUNDROBIN	Which service is at the top of a list of services. After that service is selected for a connection, it moves to the bottom of the list.
LEASTRESPONSETIME	Which load balanced server currently has the quickest response time.
URLHASH	A hash of the destination URL.
DOMAINHASH	A hash of the destination domain.
DESTINATIONIPHASH	A hash of the destination IP address.
SOURCEIPHASH	A hash of the source IP address.
SRCIPDESTIPHASH	A hash of the source and destination IP addresses.
CALLIDHASH	A hash of the call ID in the SIP header.
LEASTBANDWIDTH	Which service currently has the fewest bandwidth constraints.
LEASTPACKETS	Which service currently is receiving the fewest packets.
CUSTOMLOAD	Data from a load monitor.
TOKEN	The configured token.

Depending on the protocol of the service that it is load balancing, the NetScaler appliance sets up each connection between client and server to last for a different time interval. This is called load balancing granularity, of which are three types: request-based, connection-based, and time-based granularity. The following table describes each type of granularity and when each is used.

Granularity	Types of Load Balanced Service	Specifies
Request -based	HTTP or HTTPS	A new service is chosen for each HTTP request, independent of TCP connections. As with all HTTP requests, after the Web server fulfils the request, the connection is closed.
Connection-based	TCP and TCP-based protocols other than HTTP	A service is chosen for every new TCP connection. The connection persists until terminated by either the service or the client.
Time-based	UDP and other IP protocols	A new service is chosen for each UDP packet. Upon selection of a service, a session is created between the service and a client for a specified period of time. When the time expires, the session is deleted and a new service is chosen for any additional packets, even if those packets come from the same client.

During startup of a virtual server, or whenever the state of a virtual server changes, the virtual server can initially use the round robin method to distribute the client requests among the physical servers. This type of distribution, referred to as *startup round robin,* helps prevent unnecessary load on a single server as the initial requests are served. After using the round robin method at the startup, the virtual server switches to the load balancing method specified on the virtual server.

The Startup RR Factor works in the following manner:

- If the Startup RR Factor is set to zero, the NetScaler switches to the specified load balancing method depending on the request rate.
- If the Startup RR Factor is any number other than zero, NetScaler uses the round robin method for the specified number of requests before switching to the specified load balancing method.
- By default, the Startup RR Factor is set to zero.

Note: You cannot set the startup RR Factor for an individual virtual server. The value you specify applies to all the virtual servers on the NetScaler appliance.

To set the startup round-robin factor by using the NetScaler command line

At the NetScaler command prompt, type:

set lb parameter **-startupRRFactor** <positive_integer>

Example

```
set lb parameter -startupRRFactor 25000
```

Parameter for setting the startup round-robin factor

startupRRFactor

The number of requests for which the virtual server is to apply the round robin load balancing method (slow start mode) before switching to the load balancing method configured on the virtual server. Minimum value: 0, Maximum value: 4294967295. Default: 0.

To set the startup round-robin factor by using the NetScaler configuration utility

1. In the navigation pane, click **Load Balancing**.
2. Under **Settings**, click **Configure Load Balancing Parameters**.
3. In the **Configure Load Balancing Parameters** dialog box, for **Startup RR Factor** type a value.
4. Click **OK**.

The Least Connection Method

When a virtual server is configured to use the least connection load balancing algorithm (or method), it selects the service with the fewest active connections. This is the default method, because, in most circumstances, it provides the best performance.

For TCP, HTTP, HTTPS, and SSL_TCP services, the NetScaler appliance includes the following connection types in its list of existing connections:

- **Active connections to a service.** Connections representing requests that a client has sent to the virtual server and that the virtual server has forwarded to a service. For HTTP and HTTPS services, active connections represent only those HTTP or HTTPS requests that have not yet received a response.
- **Waiting connections in the surge queue.** Any connections to the virtual server that are waiting in a surge queue and have not yet been forwarded to a service. Connections can build up in the surge queue at any time, for any of the following reasons:
  - Your services have connection limits, and all services in your load balancing configuration are at that limit.
  - The surge protection feature is configured and has been activated by a surge in requests to the virtual server.
  - The load-balanced server has reached an internal limit and therefore does not open any new connections. (For example, an Apache server's connection limit is reached.)

  When a virtual server uses the least connection method, it considers the waiting connections as belonging to the specific service. Therefore, it does not open new connections to those services.

For UDP services, the connections that the least connection algorithm considers include all sessions between the client and a service. These sessions are logical, time-based entities. When the first UDP packet in a session arrives, the NetScaler appliance creates a session between the source IP address and port and the destination IP address and port.

For Real-Time Streaming Protocol (RTSP) connections, the NetScaler appliance uses the number of active control connections to determine the lowest number of connections to an RTSP service.

The following example shows how a virtual server selects a service for load balancing by using the least connection method. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions.
- Service-HTTP-2 is handling 15 active transactions.
- Service-HTTP-3 is not handling any active transactions.

The following diagram illustrates how the NetScaler appliance forwards incoming requests when using the least connection method.

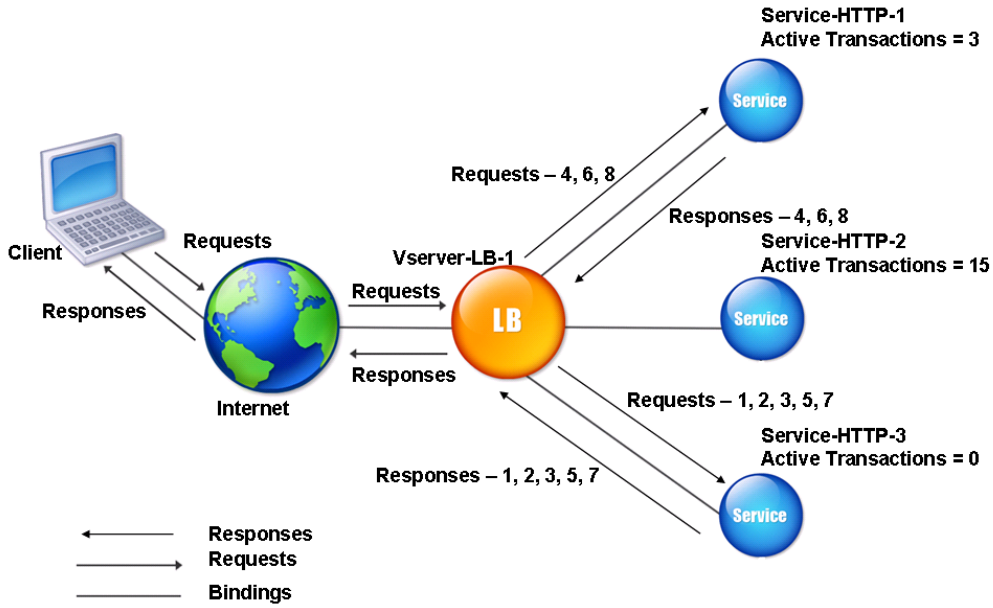


Figure 1. Mechanism of the Least Connections Load Balancing Method

In this diagram, the virtual server selects the service for each incoming connection by choosing the server with the fewest active transactions.

Connections are forwarded as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  Note: The service with no active transaction is selected first.

- Service-HTTP-3 receives the second and third requests because the service has the next least number of active transactions.
- Service-HTTP-1 receives the fourth request Because Service-HTTP-1 and Service-HTTP-3 have same number of active transactions, the virtual server uses the round robin method to chose between them.
- Service-HTTP-3 receives the fifth request.
- Service-HTTP-1 receives the sixth request, and so on, until both Service-HTTP-1 and Service-HTTP-3 are handling the same number of requests as Service-HTTP-2. At that time, the NetScaler appliance starts forwarding requests to Service-HTTP-2 when it is the least loaded service or its turn comes up in the round robin queue.

Note: If connections to Service-HTTP-2 close, it might get new connections before each of the other two services has 15 active transactions.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Incoming Connection	Service Selected	Current Number of Active Connections	Remarks
Request-1	Service-HTTP-3 (N = 0)	1	Service-HTTP-3 has the fewest active connections.
Request-2	Service-HTTP-3 (N = 1)	2	
Request-3	Service-HTTP-3 (N = 2)	3	
Request-4	Service-HTTP-1 (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.
Request-5	Service-HTTP-3 (N = 3)	4	
Request-6	Service-HTTP-1 (N = 4)	5	
Request-7	Service-HTTP-3 (N = 4)	5	
Request-8	Service-HTTP-1 (N = 5)	6	
Service-HTTP-2 is selected for load balancing when it completes its active transactions and the current connections to it close, or when the other services (Service-HTTP-1 and Service-HTTP-3) have 15 or more connections each.			

The NetScaler appliance can also use the least connection method when weights are assigned to services. It selects a service by using the value (Nw) of the following expression:

*Nw = (Number of active transactions) * (10000 / weight)*

The following example shows how the NetScaler appliance selects a service for load balancing by using the least connection method when weights are assigned to services. In the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. Connections are forwarded as follows:

- Service-HTTP-3 receives the first because the service is not handling any active transactions.

Note: If services are not handling any active transactions, the NetScaler appliance uses the round robin method regardless of the weights assigned to each of the services.

- Service-HTTP-3 receives the second, third, fourth, fifth, sixth, and seventh requests because the service has lowest Nw value.
- Service-HTTP-1 receives the eighth request. Because Service-HTTP-1 and Service-HTTP-3 now have same Nw value, the NetScaler performs load balancing in a round robin manner. Therefore, Service-HTTP-3 receives the ninth request.

The following table explains how connections are distributed on the three-service load balancing setup that is described above.

Request Received	Service Selected	Current Nw (Number of active transactions) * (10000 / weight) value	Remarks
Request-1	Service-HTTP-3 (Nw = 0)	Nw = 2500	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 2500)	Nw = 5000	
Request-3	Service-HTTP-3 (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3 (Nw = 7500)	Nw = 10000	
Request-5	Service-HTTP-3 (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3 (Nw = 12500)	Nw = 15000	
Request-7	Service-HTTP-1 (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw values
Request-8	Service-HTTP-3 (Nw = 15000)	Nw = 17500	
Service-HTTP-2 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-1 and Service-HTTP-3) is equal to 50000.			

The following diagram illustrates how the NetScaler appliance uses the least connection method when weights are assigned to the services.

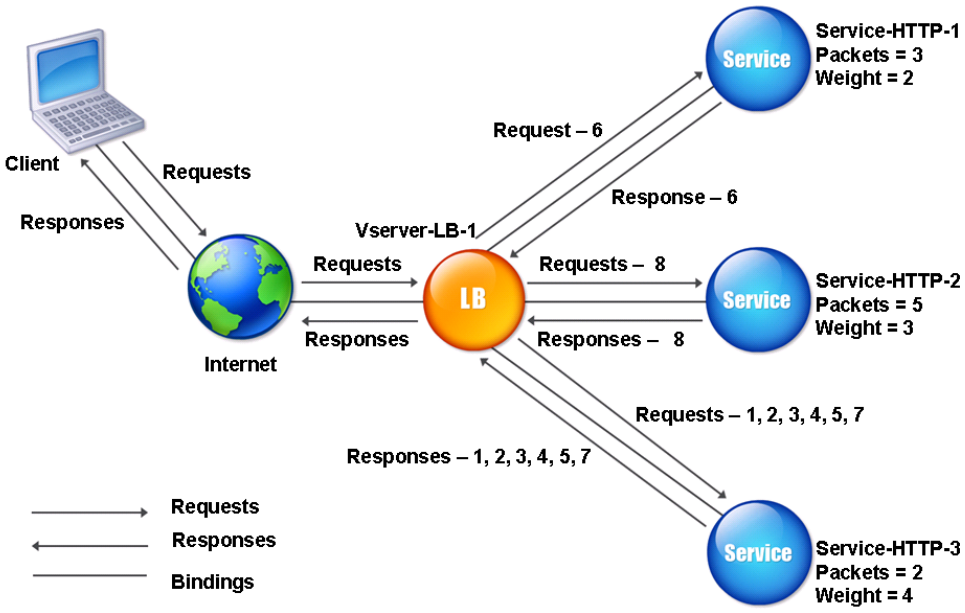


Figure 2. Mechanism of the Least Connections Load Balancing Method when Weights are Assigned

To configure the least connection method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Round Robin Method

When a load balancing virtual server is configured to use the round robin method, it continuously rotates a list of the services that are bound to it. When the virtual server receives a request, it assigns the connection to the first service in the list, and then moves that service to the bottom of the list.

The following diagram illustrates how the NetScaler appliance uses the round robin method with a load balancing setup that contains three load-balanced servers and their associated services.

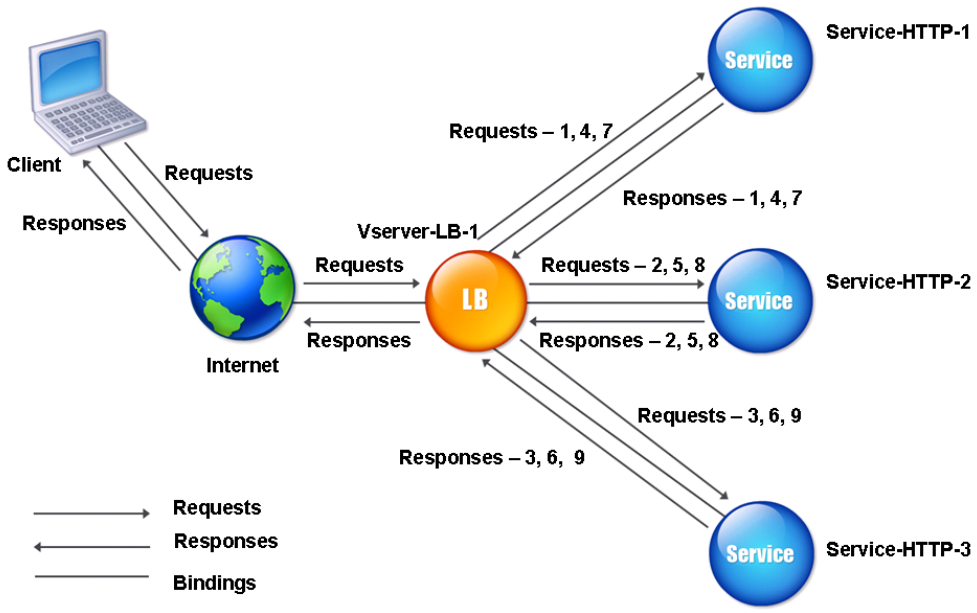


Figure 1. How the Round Robin Load Balancing Method Works

If you assign a different weight to each service, the NetScaler appliance performs weighted round robin distribution of incoming connections. It does this by skipping the lower-weighted services at appropriate intervals.

For example, assume that you have a load balancing setup with three services. You set Service-HTTP-1 to a weight of 2, Service-HTTP-2 to a weight of 3, and Service-HTTP-3 to a weight of 4. The services are bound to Vserver-LB-1, which is configured to use the round robin method. With this setup, incoming requests are delivered as follows:

- Service-HTTP-1 receives the first request.
- Service-HTTP-2 receives the second request.

- Service-HTTP-3 receives the third request.
- Service-HTTP-1 receives the fourth request.
- Service-HTTP-2 receives the fifth request.
- Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh request.
- Service-HTTP-3 receives both the eighth and the ninth requests.

Note: You can also configure weights on services to prevent multiple services from using the same server and overloading the server.

A new cycle then begins, using the same pattern.

The following diagram illustrates the weighted round robin method.

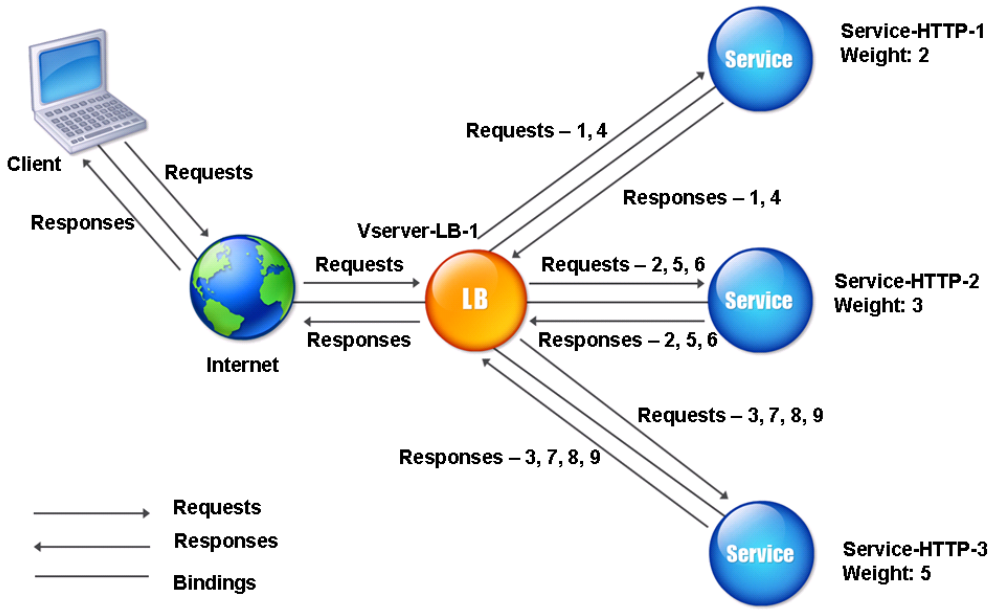


Figure 2. How the Round Robin Load Balancing Method Works with Weighted Services

To configure the round robin method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Least Response Time Method

When the load balancing virtual server is configured to use the least response time method, it selects the service with the fewest active connections and the lowest average response time. You can configure this method for HTTP and Secure Sockets Layer (SSL) services only. The response time (also called Time to First Byte, or TTFB) is the time interval between sending a request packet to a service and receiving the first response packet from the service. The NetScaler appliance uses response code 200 to calculate TTFB.

The following example shows how a virtual server selects a service for load balancing by using the least response time method. Consider the following three services:

- Service-HTTP-1 is handling three active transactions and TTFB is two seconds.
- Service-HTTP-2 is handling seven active transactions and TTFB is one second.
- Service-HTTP-3 is not handling any active transactions and TTFB is two seconds.

The following diagram illustrates how the NetScaler appliance uses the least response time method to forward the connections.

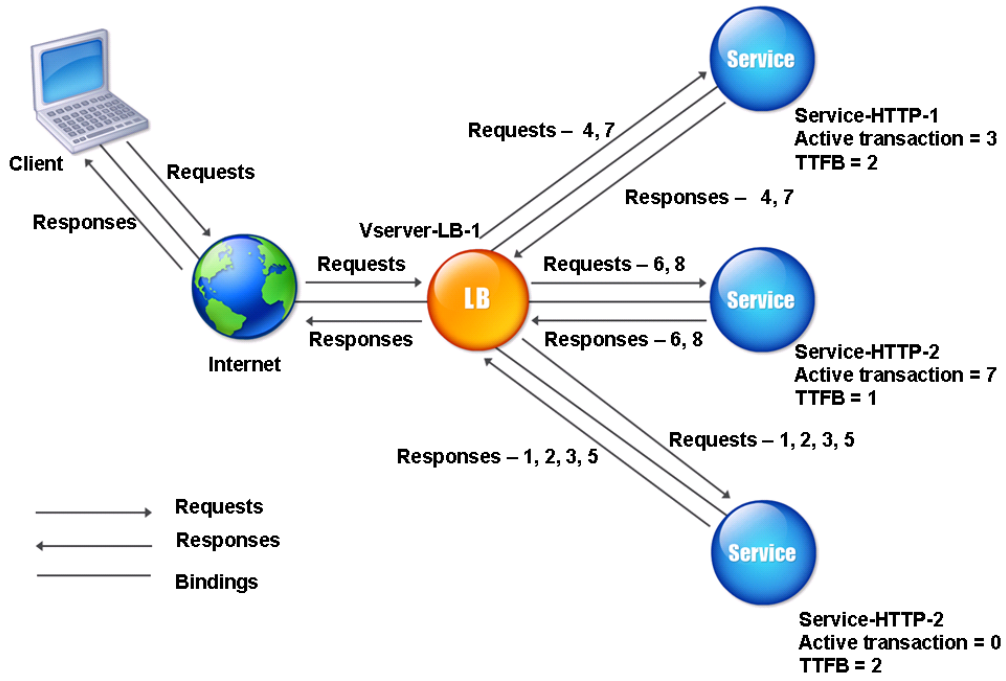


Figure 1. How the Least Response Time Load Balancing Method Works

The virtual server selects a service by multiplying the number of active transactions by the TTFB for each service and then selecting the service with the lowest result. For the

example shown above, the virtual server forwards requests as follows:

- Service-HTTP-3 receives the first request, because the service is not handling any active transactions.
- Service-HTTP-3 also receives the second and third requests, because the result is lowest of the three services.
- Service-HTTP-1 receives the fourth request. Since Service-HTTP-1 and Service-HTTP-3 have the same result, the NetScaler appliance chooses between them by applying the Round Robin method.
- Service-HTTP-3 receives the fifth request.
- Service-HTTP-2 receives the sixth request, because at this point it has the lowest result.
- Because Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all have the same result at this point, the NetScaler switches to the round robin method, and continues to distribute connections using that method.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-1	Service-HTTP-3 (N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3 (N = 2)	N = 4	
Request-3	Service-HTTP-3 (N = 3)	N = 6	
Request-4	Service-HTTP-1 (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-5	Service-HTTP-3 (N = 6)	N = 8	
Request-6	Service-HTTP-2 (N = 7)	N = 8	Service-HTTP-2 has the lowest N value.

Request-7	Service-HTTP-1 (N = 8)	N = 15	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-2 (N = 8)	N = 9	

The virtual server selects a service by using the value (Nw) in the following expression:

*Nw = (N) * (10000 / weight)*

Suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned weight of 3, and Service-HTTP-3 is assigned weight of 4.

The NetScaler appliance distributes requests as follows:

Service-HTTP-3 receives the first request, because it is not handling any active transactions.

If services are not handling any active transactions, the NetScaler selects them regardless of the weights assigned to them.

- Service-HTTP-3 receives the second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and therefore the highest Nw value, so the virtual server does not select it for load balancing.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 0)	Nw = 2500	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 2500	Nw = 5000	

Request-3	Service-HTTP-3 (Nw = 5000)	Nw = 15000	
Request-4	Service-HTTP-3 (Nw = 15000)	Nw = 20000	
Request-5	Service-HTTP-3 (Nw = 20000)	Nw = 25000	
Request-6	Service-HTTP-2 (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-7	Service-HTTP-3 (Nw = 25000)	Nw= 30000	Service-HTTP-3 has the lowest Nw value.
Request-8	Service-HTTP-2 (Nw = 26666.67)	Nw = 33333.34	Service-HTTP-2 has the lowest Nw value.
Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw values of other services (Service-HTTP-2 and Service-HTTP-3) are equal to 105000.			

The following diagram illustrates how the NetScaler appliance uses the least response time method when weights are assigned.

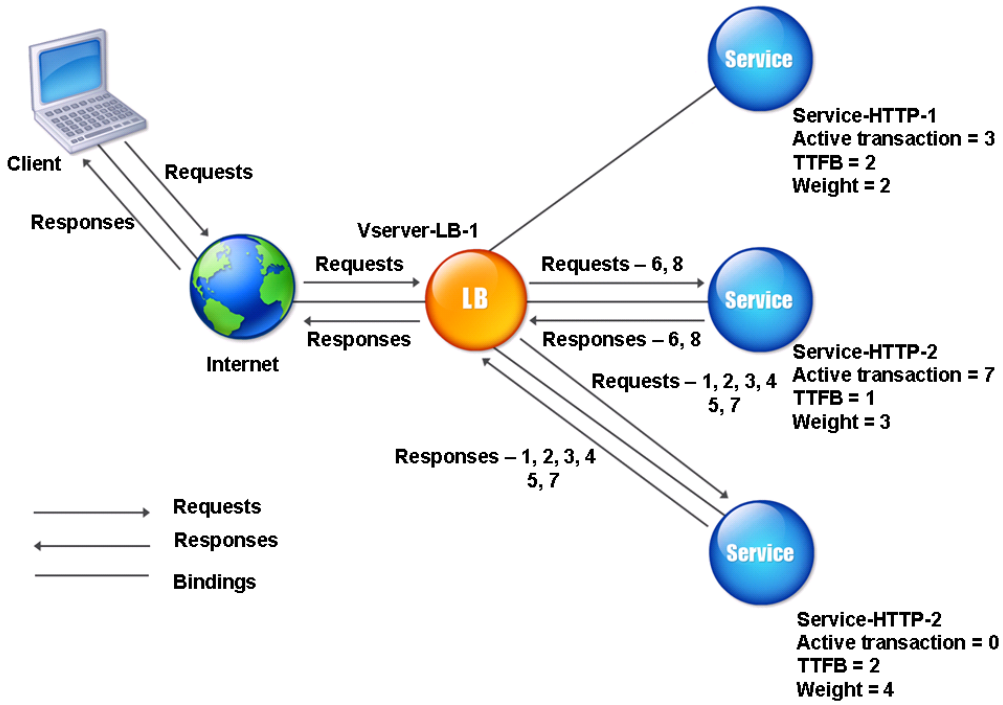


Figure 2. How the Least Response Time Load Balancing Method Works When Weights Are Assigned

To configure the least response time method, see Configuring a Load Balancing Method that Does Not Include a Policy.

When a load balancing virtual server is configured to use the least response time method with monitors, it uses the existing monitoring infrastructure to select the service with the smallest number of active transactions and the fastest average response time. Before you use the least response time method with monitoring, you must bind application-specific monitors to each service and enable least response time method mode on these monitors. The NetScaler appliance then makes load balancing decisions based on the response times it calculates from monitoring probes. For more information about configuring monitors, see Configuring Monitors in a Load Balancing Setup.

You can use the least response time method with monitors to select non-HTTP and non-HTTPS services. You can also use this method when several monitors are bound to a service. Each monitor determines the response time by using the protocol that it measures for the service that it is bound to.The virtual server then calculates an average response time for that service by averaging the results.

The following table summarizes how response times are calculated for various monitors.

Monitor	Response Time Calculation
PING	Time difference between the ICMP ECHO request and the ICMP ECHO response.
TCP	Time difference between the SYN request and the SYN+ACK response.
HTTP	Time difference between the HTTP request (after the TCP connection is established) and the HTTP response.
TCP-ECV	Time difference between the time the data send string is sent and the data receive string is returned. A tcp-ecv monitor without the send and receive strings is considered to have an incorrect configuration.
HTTP-ECV	Time difference between the HTTP request and the HTTP response.
UDP-ECV	Time difference between the UDP send string and the UDP receive string. A udp-ecv monitor without the receive string is considered to have an incorrect configuration.
DNS	Time difference between a DNS query and the DNS response.
TCPS	Time difference between a SYN request and the SSL handshake completion.
FTP	Time difference between the sending of the user name and the completion of user authentication.

HTTPS (monitors HTTPS requests)	Time difference is same as for the HTTP monitor.
HTTPS-ECV (monitors HTTPS requests)	Time difference is same as for the HTTP-ECV monitor
USER	Time difference between the time when a request is sent to the dispatcher and the time when the dispatcher response is received.

The following example shows how the NetScaler appliance selects a service for load balancing by using the least response time method with monitors. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions and the response time is five seconds.
- Service-HTTP-2 is handling 7 active transactions and the response time is one second.
- Service-HTTP-3 is not handling any active transactions and the response time is two seconds.

The following diagram illustrates the process that the NetScaler appliance follows when it forwards requests.

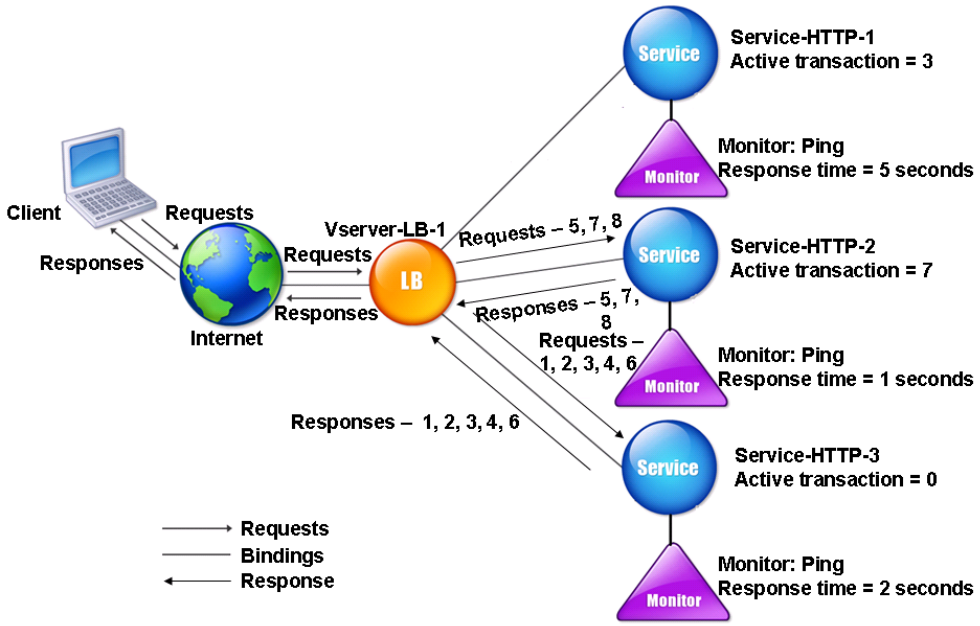


Figure 3. How the Least Response Time Load Balancing Method Works When Using Monitors

The virtual server selects a service by using the value (N) in the following expression:

*N = Number of active transactions * Response time that is determined by the monitor*

The virtual server delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service is not handling any active transaction.
- Service-HTTP-3 receives the second, third, and fourth requests, because this service has the lowest N value.
- Service-HTTP-2 receives the fifth request, because this service has the lowest N value.
- Since both Service-HTTP-2 and Service-HTTP-3 currently have the same N value, the NetScaler appliance switches to the round robin method Therefore, Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh and eighth requests, because this service has the lowest N value.

Service-HTTP-1 is not considered for load balancing, because it is more heavily loaded (has the highest N value) when compared to the other two services. However, if Service-HTTP-1 completes its active transactions, the NetScaler appliance again considers that service for load balancing.

The following table summarizes how N is calculated for the services.

Request Received	Service Selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-3 (N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3 (N = 2)	N = 4	
Request-3	Service-HTTP-3 (N = 4)	N = 6	
Request-4	Service-HTTP-3 (N = 6)	N = 8	
Request-5	Service-HTTP-2 (N = 7)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-6	Service-HTTP-3 (N = 8)	N = 10	
Request-7	Service-HTTP-2 (N = 8)	N = 9	Service-HTTP-2 has the lowest N value.

Request-8	Service-HTTP-1 (N = 9)	N = 10	
Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when the N value of the other services (Service-HTTP-2 and Service-HTTP-3) is equal to 15.			

The NetScaler appliance also performs load balancing by using the number of active transactions, response time, and weights if different weights are assigned to services. The NetScaler appliance selects the service by using the value (Nw) in the following expression:

*Nw = (N) * (10000 / weight)*

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4.

The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.
- Service-HTTP-3 receives the second, third, and fourth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the fifth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the seventh and the eighth requests, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and the highest Nw value, so the NetScaler appliance does not select it for load balancing.

The following table summarizes how Nw is calculated for various monitors.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 0)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 5000	Nw = 10000	
Request-3	Service-HTTP-3 (Nw = 15000)	Nw = 20000	

Request-4	Service-HTTP-3 (Nw = 20000)	Nw = 25000	
Request-5	Service-HTTP-2 (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-6	Service-HTTP-3 (Nw = 25000)	Nw = 30000	Service-HTTP-3 has the lowest Nw value.
Request-7	Service-HTTP-2 (Nw = 23333.34)	Nw= 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-8	Service-HTTP-2 (Nw =25000)	Nw = 30000	Service-HTTP-2 has the lowest Nw value.
Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw value of the other services (Service-HTTP-2 and Service-HTTP-3) is equal to 75000.			

The following diagram illustrates how the virtual server uses the least response time method when weights are assigned.

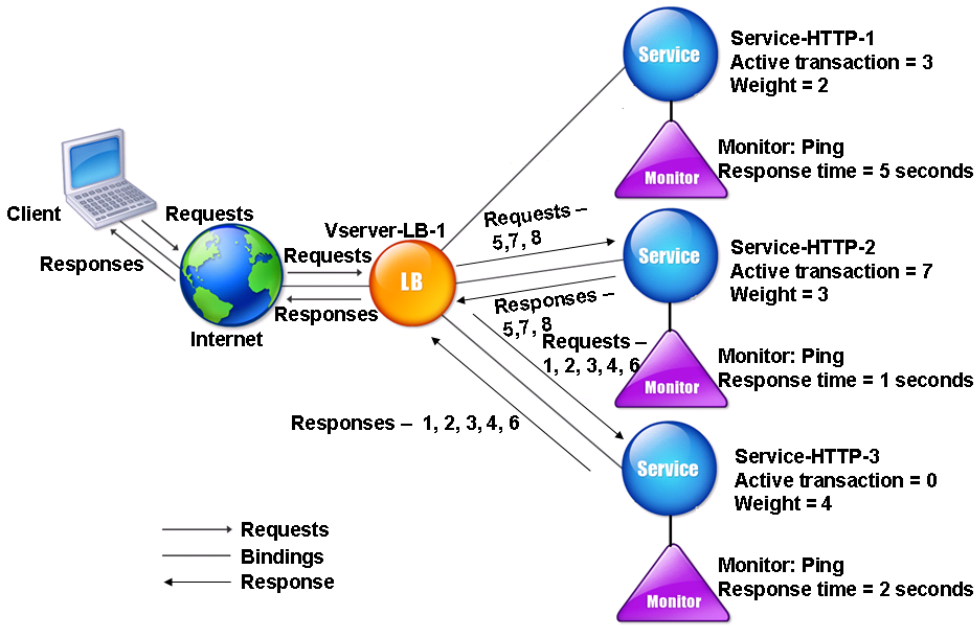


Figure 4. How the Least Response Time Load Balancing Method with Monitors Works When Weights Are Assigned

To configure the least response time method using monitors, see Configuring a Load Balancing Method that Does Not Include a Policy.

About Hashing Methods

Load balancing methods based on hashes of certain connection information or header information constitute the majority of the NetScaler appliance's load balancing methods. Hashes are shorter and easier to use than the information that they are based on, while retaining enough information to ensure that no two different pieces of information generate the same hash and are therefore confused with one another.

You can use the hashing load balancing methods in an environment where a cache serves a wide range of content from the Internet or specified origin servers. Caching requests reduces request and response latency, and ensures better resource (CPU) utilization, making caching popular on heavily used Web sites and application servers. Since these sites also benefit from load balancing, hashing load balancing methods are widely useful.

The NetScaler provides the following hashing methods:

- URL hash method
- Domain hash method
- Destination IP hash method
- Source IP hash method
- Source IP Destination IP hash method
- Source IP Source Port hash method
- Call ID hash method
- Token method

These hashing algorithms ensure minimal disruption when services are added to or deleted from your load balancing setup. Most of them calculate two hash values:

- A hash of the service's IP address and port.
- A hash of the incoming URL, the domain name, the source IP address, the destination IP address, or the source and destination IP addresses, depending on the configured hash method.

The NetScaler appliance then generates a new hash value by using both of those hash values. Finally, it forwards the request to the service with highest hash value. As the appliance computes a hash value for each request and selects the service that will process the request, it populates a cache. Subsequent requests with the same hash value are sent to the same service. The following flow chart illustrates this process.

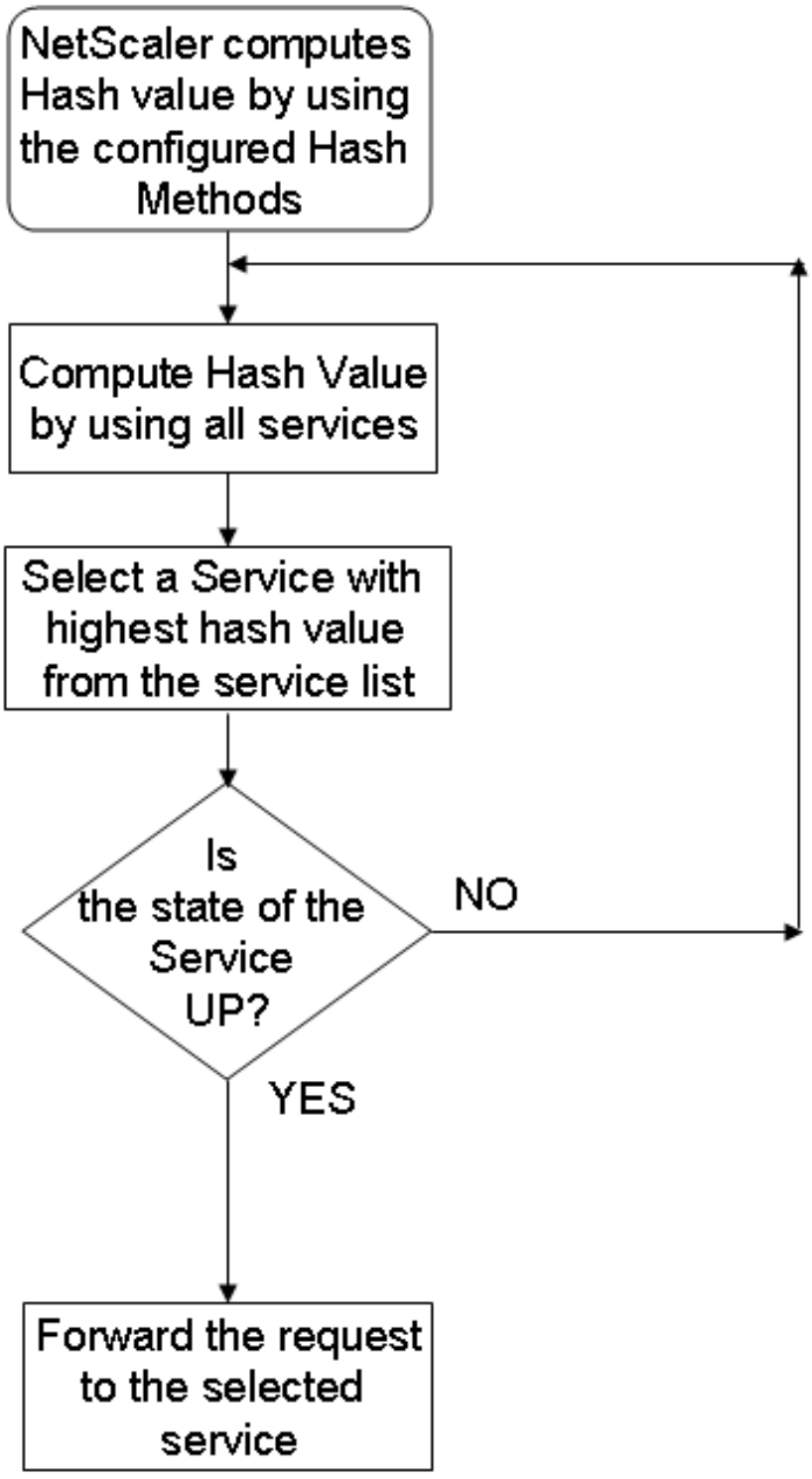


Figure 1. How the Hashing Methods Distribute Requests

Hashing methods can be applied to IPv4 and IPv6 addresses.

Consider a scenario where three services (Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3) are bound to a virtual server, any hash method is configured, and the hash value is Hash1. When the configured services are UP, the request is sent to Service-HTTP-1. If Service-HTTP-1 is down, the NetScaler appliance calculates the hash value for the last log

of the number of services. The NetScaler then selects the service with the highest hash value, such as Service-HTTP-2. The following diagram illustrates this process.

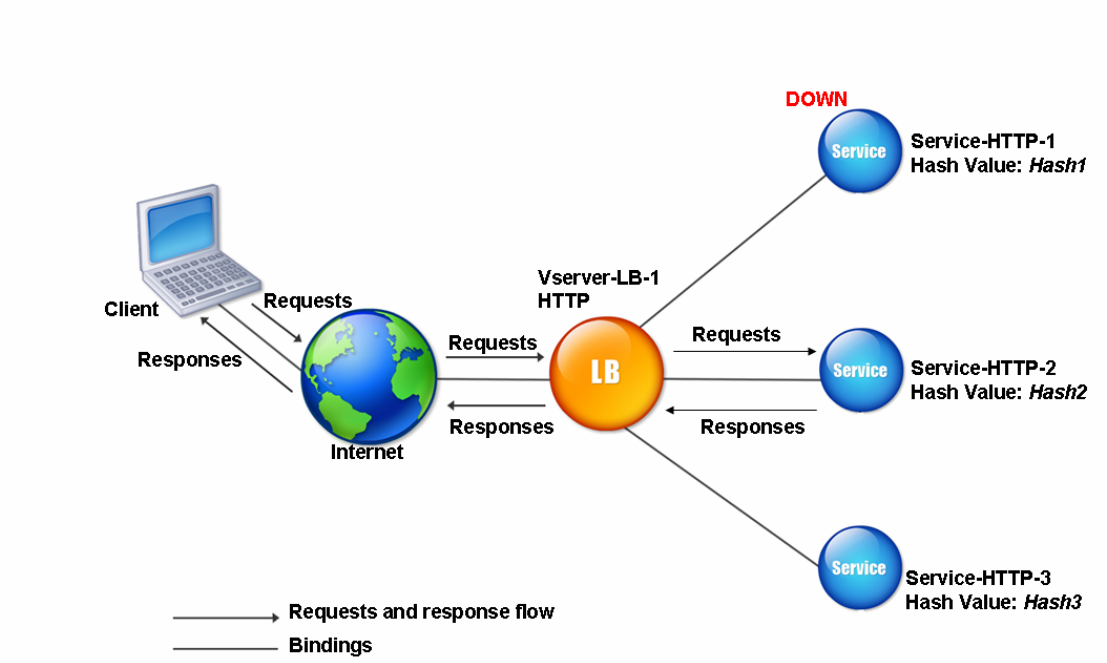


Figure 2. Entity Model for Hashing Methods

Note: If the NetScaler appliance fails to select a service by using a hashing method, it defaults to the least connection method to select a service for the incoming request. You should adjust server pools by removing services during periods of low traffic to enable the caches to repopulate without affecting performance on your load balancing setup.

The URL Hash Method

When you configure the NetScaler appliance to use the URL hash method for load balancing the services, for selecting a service, the NetScaler generates a hash value of the HTTP URL present in the incoming request. If the service selected by the hash value is DOWN, the algorithm has a method to select another service from the list of active services. The NetScaler caches the hashed value of the URL, and when it receives subsequent requests that use the same URL, it forwards them to the same service. If the NetScaler cannot parse an incoming request, it uses the round robin method for load balancing instead of the URL hash method.

For generating the hash value, NetScaler uses a specific algorithm and considers a part of the URL. By default, the NetScaler considers the first 80 bytes of the URL. If the URL is of less than 80 bytes, the complete URL is used. You can specify a different length. The hash length can be from 1 to 4096 bytes. Generally, if long URLs are used where only a small number of characters are different, it is a good idea to make the hash length as high as possible to try to ensure a more even load distribution.

Consider a scenario where three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3, are bound to a virtual server, and the load balancing method configured on the virtual server is the URL hash method. The virtual server receives a request and the hash value of the URL is U1. NetScaler selects Service-HTTP-1. If Service-HTTP-1 is DOWN, the NetScaler selects Service-HTTP-2.

The following diagram illustrates this process.

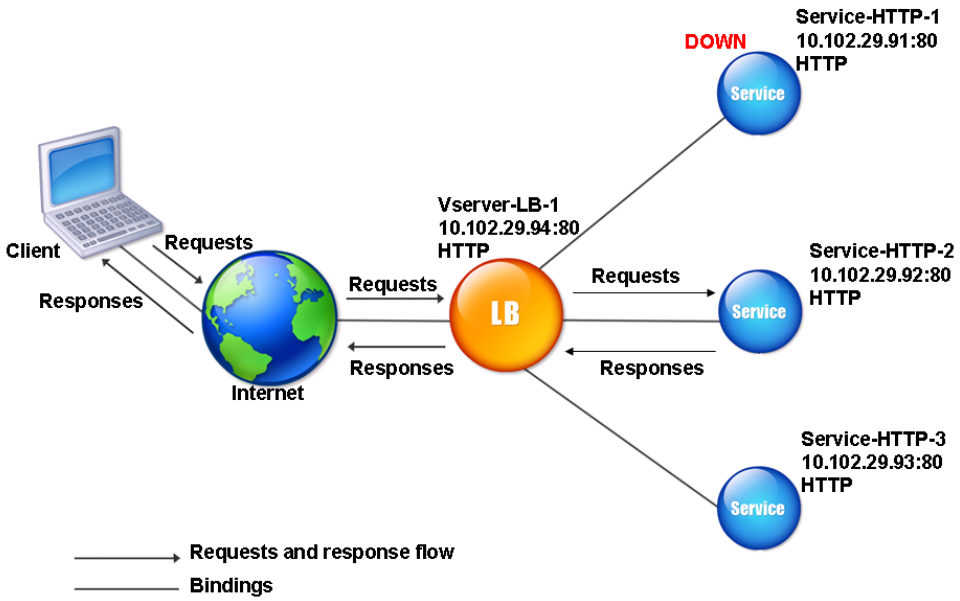


Figure 1. How URL Hashing Operates

If both Service-HTTP-1 and Service-HTTP-2 are DOWN, NetScaler sends requests with hash value U1 to Service-HTTP-3.

If Service-HTTP-1 and Service-HTTP-2 are down, requests that generate the hash URL1 are sent to Service-HTTP-3. If these services are UP, requests that generate the hash URL1 are distributed in the following manner:

- If the Service-HTTP-2 is up, the request is sent to Service-HTTP-2.
- If the Service-HTTP-1 is up, the request is sent to Service-HTTP-1.
- If Service-HTTP-1 and Service-HTTP-2 are up at the same time, the request is sent to Service-HTTP-1.

To configure the URL hash method, see Configuring a Load Balancing Method that Does Not Include a Policy. Select the load balancing method as URL Hash, and set the hash length to the number of bytes to be used for generating the hash value.

The Domain Hash Method

A load balancing virtual server configured to use the domain hash method uses the hashed value of the domain name in the HTTP request to select a service. The domain name is taken from either the incoming URL or the Host header of the HTTP request. If the domain name appears in both the URL and the Host header, the NetScaler gives preference to the URL.

If you configure domain name hashing, and an incoming HTTP request does not contain a domain name, the NetScaler appliance defaults to the round robin method for that request.

The hash-value calculation uses the name length or hash length value, whichever is smaller. By default, the NetScaler appliance calculates the hash value from the first 80 bytes of the domain name. To specify a different number of bytes in the domain name when calculating the hash value, you can set the hashLength parameter (Hash Length in the configuration utility) to a value of from 1 to 4096 (bytes).

To configure the domain hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Destination IP Hash Method

A load balancing virtual server configured to use the destination IP hash method uses the hashed value of the destination IP address to select a server. You can mask the destination IP address to specify which part of it to use in the hash-value calculation, so that requests that are from different networks but destined for the same subnet are all directed to the same server. This method supports IPv4 and IPv6-based destination servers.

This load balancing method is appropriate for use with the cache redirection feature. For more information about the cache redirection feature, see Cache Redirection.

To configure the destination IP hash method for an IPv4 destination server, you set the netMask parameter. To configure this method for an IPv6 destination server, you use the **v6NetMaskLen** parameter. In the configuration utility, text boxes for setting these parameters appear when you select the Destination IP Hash method.

To configure the destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Hash Method

A load balancing virtual server configured to use the source IP hash method uses the hashed value of the client IPv4 or IPv6 address to select a service. To direct all requests from source IP addresses that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the **netMask** parameter. For IPv6 addresses, use the **v6NetMaskLength** parameter.

To configure the source IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Destination IP Hash Method

A load balancing virtual server configured to use the source IP destination IP hash method uses the hashed value of the source and destination IP addresses (either IPv4 or IPv6) to select a service. Hashing is symmetric; the hash-value is the same regardless of the order of the source and destination IPs. This ensures that all packets flowing from a particular client to the same destination are directed to the same server.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the **netMask** parameter. For IPv6 addresses, use the **v6NetMaskLength** parameter.

To configure the source IP destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Source Port Hash Method

A load balancing virtual server configured to use the source IP source port hash method uses the hash value of the source IP (either IPv4 or IPv6) and source port to select a service. This ensures that all packets on a particular connection are directed to the same service.

This method is used in connection mirroring and firewall load balancing. For more information about connection mirroring, see Connection Failover.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the **netMask** parameter. For IPv6 addresses, use the **v6NetMaskLength** parameter.

To configure the source IP source port hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Call ID Hash Method

A load balancing virtual server configured to use the call ID hash method uses the hash value of the call ID in the SIP header to select a service. Packets for a particular SIP session are therefore always directed to the same proxy server.

This method is applicable to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure the call ID hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Least Bandwidth Method

A load balancing virtual server configured to use the least bandwidth method selects the service that is currently serving the least amount of traffic, measured in megabits per second (Mbps). The following example shows how the virtual server selects a service for load balancing by using the least bandwidth method.

Consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has 3 Mbps bandwidth.
- Service-HTTP-2 has 5 Mbps bandwidth.
- Service-HTTP-3 has 2 Mbps bandwidth.

The following diagram illustrates how the virtual server uses the least bandwidth method to forward requests to the three services.

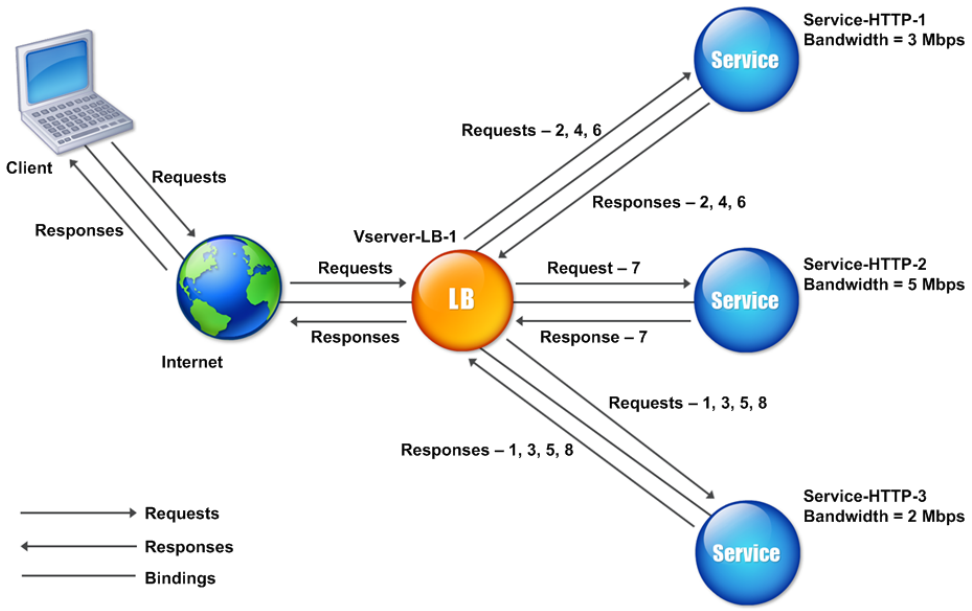


Figure 1. How the Least Bandwidth Load Balancing Method Works

The virtual server selects the service by using the bandwidth value (N), which is the sum of the number of bytes transmitted and received over the previous 14 seconds. If each request requires 1 Mbps bandwidth, the NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Since Service-HTTP-1 and Service-HTTP-3 now have same N value, the virtual server switches to the round robin method for these servers, alternating between them. Service-HTTP-1 receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 now all have same N value, the virtual server includes Service-HTTP-2 in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

Request Received	Service Selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-3 (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1 (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3 (N = 3)	N = 4	
Request-4	Service-HTTP-1 (N = 4)	N = 5	
Request-5	Service-HTTP-3 (N = 4)	N = 5	
Request-6	Service-HTTP-1 (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2 (N = 5)	N = 6	
Request-8	Service-HTTP-3 (N = 5)	N = 6	

Note: If you enable the RTSP NAT option on the virtual server, the NetScaler appliance uses the number of data and control bytes exchanged to determine the bandwidth usage for RTSP services. For more information about RTSP NAT option, see Managing RTSP Connections.

The NetScaler appliance also performs load balancing by using the bandwidth and weights if different weights are assigned to the services. It selects a service by using the value (Nw) in the following expression:

*Nw = (N) * (10000 / weight)*

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 5000)	Nw = 7500	
Request-3	Service-HTTP-3 (Nw = 7500)	Nw = 10000	
Request-4	Service-HTTP-3 (Nw = 10000)	Nw = 12500	
Request-5	Service-HTTP-3 (Nw = 12500)	Nw = 15000	
Request-6	Service-HTTP-1 (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3 (Nw = 15000)	Nw = 17500	
Request-8	Service-HTTP-2 (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least bandwidth method when weights are assigned to the services.

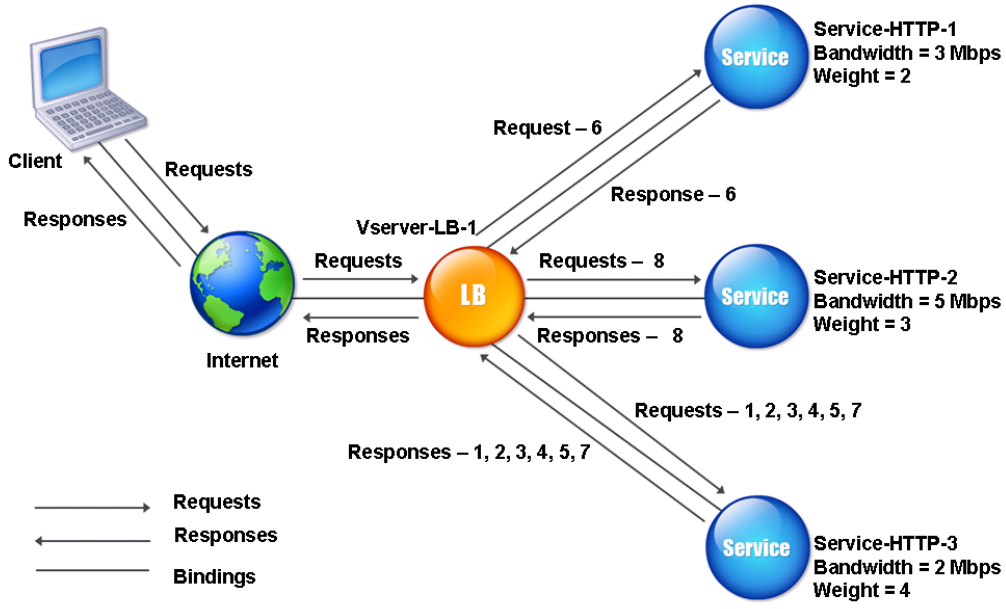


Figure 2. How the Least Bandwidth Load Balancing Method Works When Weights Are Assigned

To configure the least bandwidth method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Least Packets Method

A load balancing virtual server configured to use the least packets method selects the service that has received the fewest packets in the last 14 seconds.

For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has handled three packets in last 14 seconds.
- Service-HTTP-2 has handled five packets in last 14 seconds.
- Service-HTTP-3 has handled two packets in last 14 seconds.

The following diagram illustrates how the NetScaler appliance uses the least packets method to choose a service for each request that it receives.

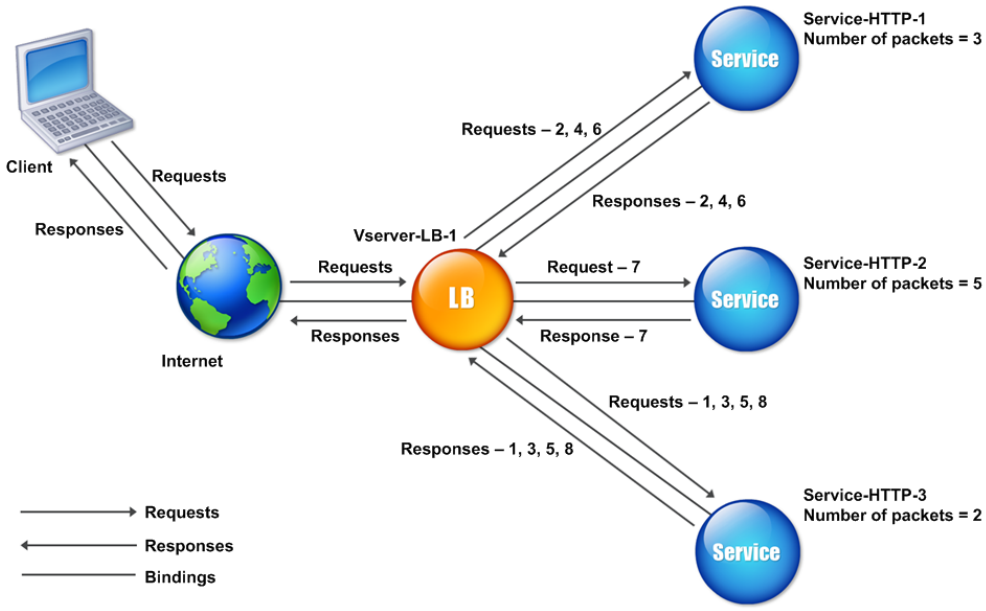


Figure 1. How the Least Packets Load Balancing Method Works

The NetScaler appliance selects a service by using the number of packets (N) transmitted and received by each service in the last 14 seconds. Using this method, it delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.

- Since Service-HTTP-1 and Service-HTTP-3 now have the same N value, the virtual server switches to the round robin method. Service-HTTP-1 therefore receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.

- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have same N value, the virtual server switches to the round robin method for Service-HTTP-2 as well, including it in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

Request Received	Service Selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-3 (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1 (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3 (N = 3)	N = 4	
Request-4	Service-HTTP-1 (N = 4)	N = 5	
Request-5	Service-HTTP-3 (N = 4)	N = 5	
Request-6	Service-HTTP-1 (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2 (N = 5)	N = 6	
Request-8	Service-HTTP-3 (N = 5)	N = 6	

Note: If you enable the RTSP NAT option on the virtual server, the NetScaler uses the number of data and control packets to calculate the number of packets for RTSP services. For more information about RTSP NAT option, see Managing RTSP Connections.

The NetScaler appliance also performs load balancing by using the number of packets and weights when a different weight is assigned to each service. It selects a service by using the value (Nw) in the following expression:

*Nw = (N) * (10000 / weight)*

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 5000)	Nw = 7500	
Request-3	Service-HTTP-3 (Nw = 7500)	Nw = 10000	
Request-4	Service-HTTP-3 (Nw = 10000)	Nw = 12500	
Request-5	Service-HTTP-3 (Nw = 12500)	Nw = 15000	
Request-6	Service-HTTP-1 (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3 (Nw = 15000)	Nw = 17500	
Request-8	Service-HTTP-2 (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least packets method when weights are assigned.

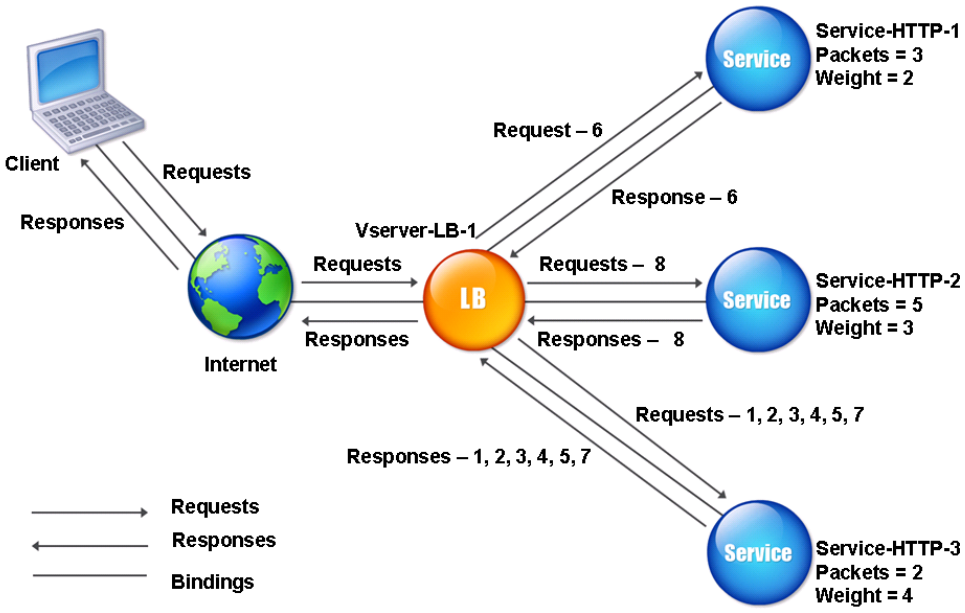


Figure 2. How the Least Packets Method Works When Weights Are Assigned

To configure the least packets method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Custom Load Method

Custom load balancing is performed on server parameters such as CPU usage, memory, and response time. When using the custom load method, the NetScaler appliance usually selects a service that is not handling any active transactions. If all of the services in the load balancing setup are handling active transactions, the appliance selects the service with the smallest load. A special type of monitor, known as a load monitor, calculates the load on each service in the network. The load monitors do not mark the state of a service, but they do take services out of the load balancing decision when those services are not UP.

For more information about load monitors, see Understanding Load Monitors. The following diagram illustrates how a load monitor operates.

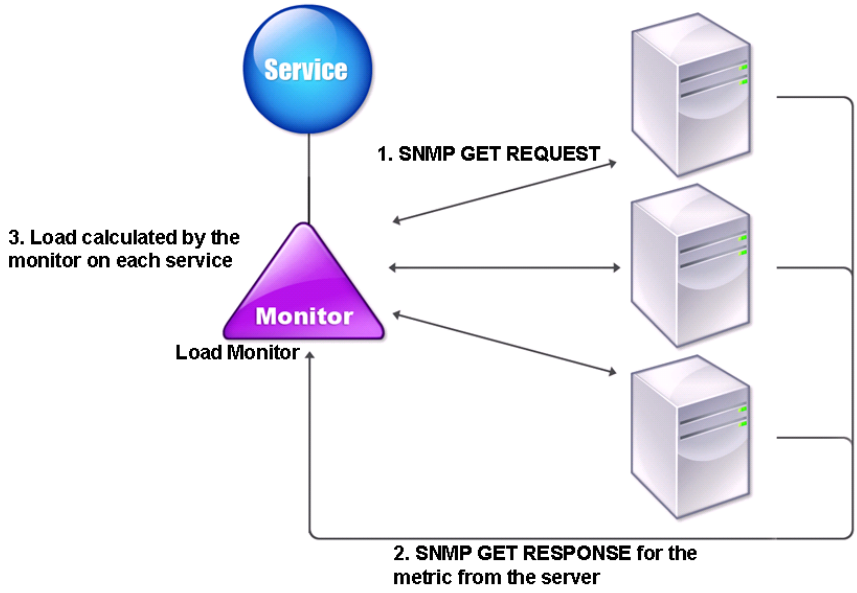


Figure 1. How Load Monitors Operate

The load monitor uses Simple Network Management Protocol (SNMP) probes to calculate load on each service by sending an SNMP GET request to the service. This request contains one or more object IDs (OIDs). The service responds with an SNMP GET response, with metrics corresponding to the SNMP OIDs. The load monitor uses the response metrics, described below, to calculate the load on the service.

The load monitor calculates the load on a service by using the following parameters:

- Metrics values retrieved through SNMP probes that exist as tables in the NetScaler.

- Threshold value set for each metric.
- Weight assigned to each metric.

For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 is using 20 megabytes (MB) of memory.
- Service-HTTP-2 is using 70 MB of memory.
- Service-HTTP-3 is using 80 MB of memory.

The load balanced servers can export metrics such as CPU and memory usage to the services, which can in turn provide them to the load monitor. The load monitor sends an SNMP GET request containing the OIDs 1.3.6.1.4.1.5951.4.1.1.41.1.5, 1.3.6.1.4.1.5951.4.1.1.41.1.4, and 1.3.6.1.4.1.5951.4.1.1.41.1.3 to the services. The three services respond to the request. The NetScaler appliance compares the exported metrics, and then selects Service-HTTP-1 because it has more available memory. The following diagram illustrates this process.

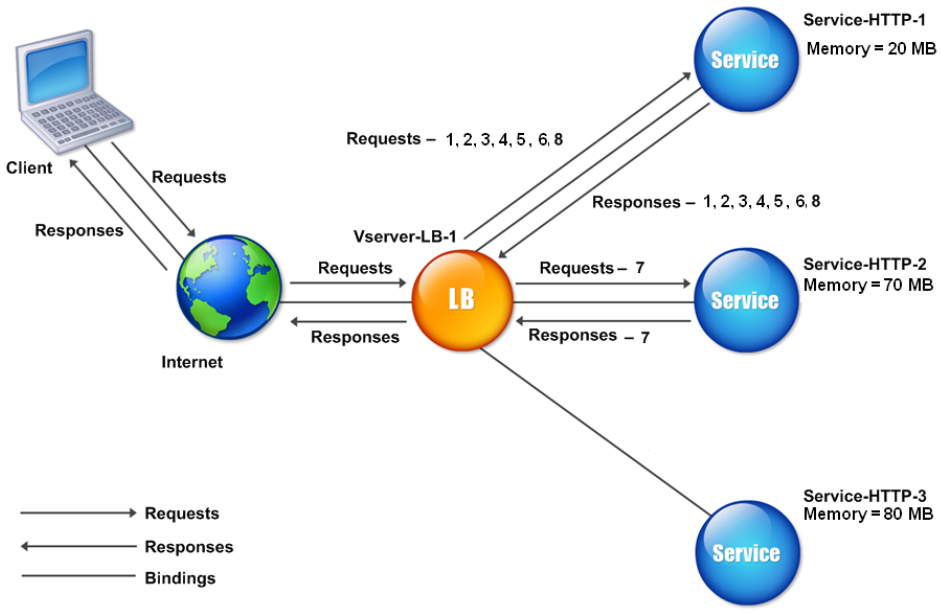


Figure 2. How the Custom Load Method Works

If each request uses 10 MB memory, the NetScaler appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, and fifth requests, because this service has the lowest N value.

- Service-HTTP-1 and Service-HTTP-2 now have the same load, so the virtual server reverts to the round robin method for these servers. Therefore, Service-HTTP-2 receives the sixth request, and Service-HTTP-1 receives the seventh request.

- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have same load, the virtual server reverts to the round robin method for Service-HTTP-3 as well. Therefore, Service-HTTP-3 receives the eighth request.

The following table summarizes how N is calculated.

Request received	Service selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-1 (N = 20)	N = 30	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1 (N = 30)	N = 40	
Request-3	Service-HTTP-1 (N = 40)	N = 50	
Request-4	Service-HTTP-1 (N = 50)	N = 60	
Request-5	Service-HTTP-1 (N = 60)	N = 70	
Request-6	Service-HTTP-1 (N = 70)	N = 80	Service-HTTP-2 and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2 (N = 70)	N = 80	
Request-8	Service-HTTP-1 (N = 80)	N = 90	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.

If different weights are assigned to the services, the custom load algorithm considers both the load on each service and the weight assigned to each service. It selects a service by using the value (Nw) in the following expression:

*Nw = (N) * (10000 / weight)*

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 4, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 2. If each request uses 10 MB memory, the NetScaler appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, fifth, sixth, seventh, and eighth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the ninth request, because this service has the lowest Nw value.

Service-HTTP-3 has the highest Nw value, and is therefore not considered for load balancing.

The following table summarizes how Nw is calculated.

Request received	Service selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-1 (Nw = 50000)	Nw = 75000	Service-HTTP-1 has the lowest Nw value.
Request-2	Service-HTTP-1 (Nw = 5000)	Nw = 100000	
Request-3	Service-HTTP-1 (Nw = 15000)	Nw = 125000	
Request-4	Service-HTTP-1 (Nw = 20000)	Nw = 150000	
Request-5	Service-HTTP-1 (Nw = 23333.34))	Nw = 175000	
Request-6	Service-HTTP-1 (Nw = 25000)	Nw = 200000	
Request-7	Service-HTTP-1 (Nw = 23333.34)	Nw = 225000	
Request-8	Service-HTTP-1 (Nw = 25000)	Nw = 250000	
Request-9	Service-HTTP-2 (Nw = 233333.34)	Nw = 266666.67	Service-HTTP-2 has the lowest Nw value.
Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-2 and Service-HTTP-3) is equal to 400,000.			

The following diagram illustrates how the NetScaler appliance uses the custom load method when weights are assigned.

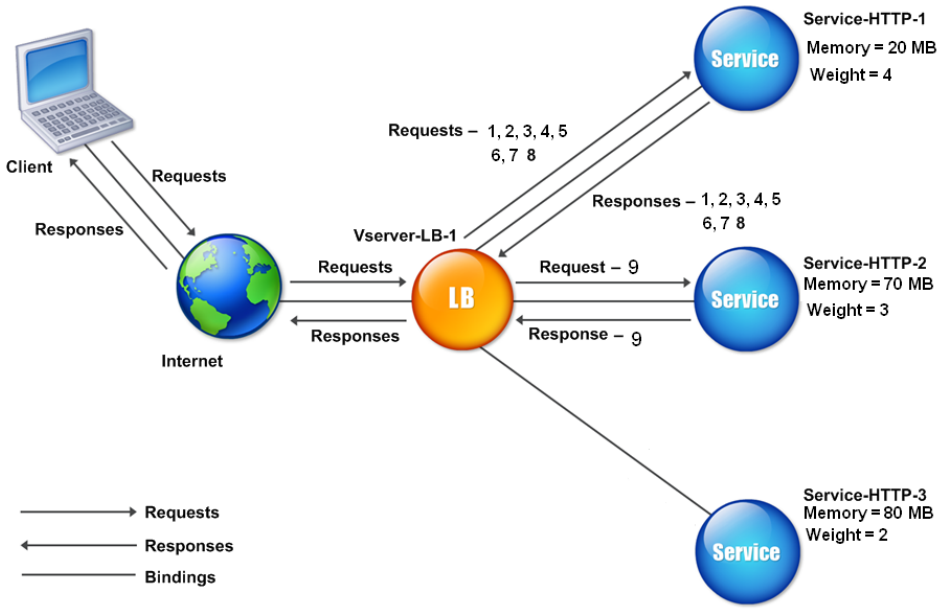


Figure 3. How the Custom Load Method Works When Weights Are Assigned

To configure the custom load method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Configuring the Token Method

A load balancing virtual server configured to use the token method bases its selection of a service on the value of a data segment extracted from the client request. The data segment is called the token. You configure the location and size of the token. For subsequent requests with the same token, the virtual server chooses the same service that handled the initial request.

This method is content aware; it operates differently for TCP, HTTP, and HTTPS connections. For HTTP or HTTPS services, the token is found in the HTTP headers, the URL, or the BODY. To locate the token, you specify or create a classic or advanced expression. For more information on classic or advanced expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

For HTTP services, the virtual server searches for the configured token in the first 24 kilobytes (KB) of the TCP payload. For non-HTTP (TCP, SSL, and SSL_TCP) services, the virtual server searches for the configured token in the first 16 packets if the total size of the 16 packets is less than 24 KB. But if the total size of the 16 packets is greater than 24 KB, the NetScaler searches for the token in the first 24 KB of payload. You can use this load balancing method across virtual servers of different types to make sure that requests presenting the same token are directed to appropriate services, regardless of the protocol used.

For example, consider a load balancing setup consisting of servers that contain Web content. You want to configure the NetScaler appliance to search for a specific string (the token) inside the URL query portion of the request. Server-1 has two services, Service-HTTP-1 and Service-TCP-1, and Server-2 has two services, Service-HTTP-2 and Service-TCP-2. The TCP services are bound to Vserver-LB-2, and the HTTP services are bound to Vserver-LB-1.

If Vserver-LB-1 receives a request with the token AA, it selects the service Service-HTTP-1 (bound to server-1) to process the request. If Vserver-LB-2 receives a different request with the same token (AA), it directs this request to the service Service-TCP-1. The following diagram illustrates this process.

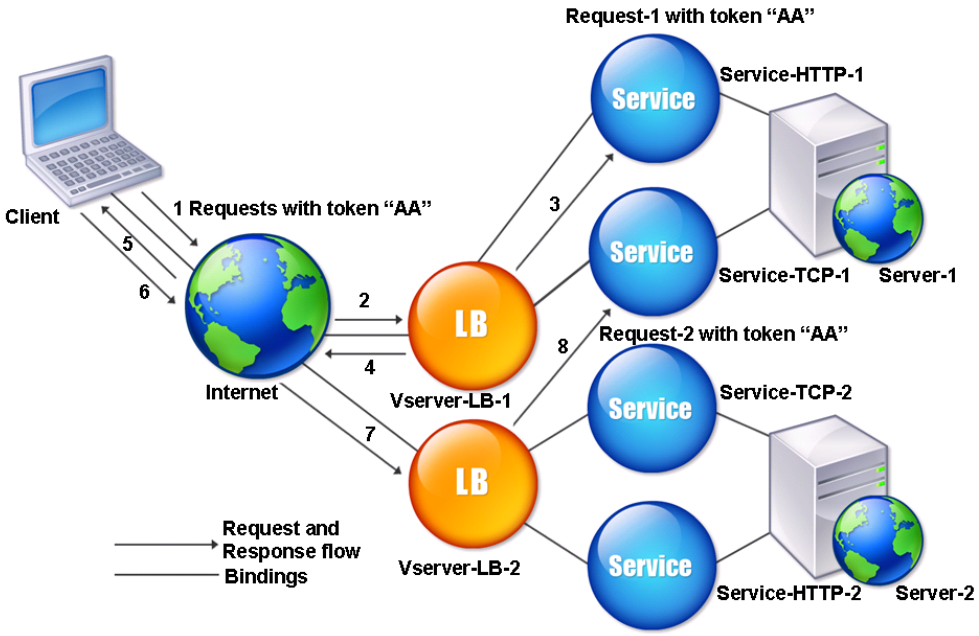


Figure 1. How the Token Method Works

To configure the Token load balancing method by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure the token load balancing method and verify the configuration:

- set lb vserver <name> -**lbMethod** TOKEN -**rule** <rule> -**datalength** <length> -**dataoffset** <offset>
- show lb vserver <name>

Example

set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -dataoffset 25

show lb vserver LB-VServer-1

Parameters for Configuring the Token Load Balancing Method

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

rule

A string. The string can be an existing rule name, or it can be an inline expression with a maximum of 256 characters.

datalength

Length of the token in bytes. This parameter is applicable to HTTP virtual servers and TCP virtual servers configured for Token load balancing. Valid values: 0-100. Default: No default.

dataoffset

Offset of the data to be taken as a token. This parameter is applicable to TCP virtual servers configured for Token load balancing. The token must be within the first 24 KB of the client TCP data. Valid values: 0-25400. Default: No default

To configure the load balancing method by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure a rule (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Method and Persistence** tab and under **LB Method**, select **Token**.
4. Click **Configure** next to the **Rule** text box.
5. In the **Create Expression** dialog box, select **Classic Syntax** or **Advanced Syntax**.
6. Under **Expression**, click **Add**.
7. In the **Add Expression** dialog box, enter an expression. For more information about expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673. For example, if you are configuring a classic expression, you can select an **Expression Type** of **General**, a Flow Type of **REQ**, a **Protocol** of **HTTP**, a **Qualifier** of **URLQUERY**, an **Operator** of **CONTAINS**, and in the **Value** text box, type **AA**.
8. Click **OK**, and then click **Close**.
9. In the **Create Expression** dialog box, click **Create**. The expression you created appears in the **Rule** text box.
10. Click **OK**.

Configuring a Load Balancing Method That Does Not Include a Policy

After you select a load balancing algorithm for your load balancing setup, you must configure the NetScaler appliance to use that algorithm. You can configure it by using the NetScaler command line or by using the configuration utility.

Note:

The token method is policy based and requires more configuration than is described here. To configure the token method, see Configuring the Token Method.

For some hash-based methods, you can mask an IP address to direct requests belonging to the same subnet to the same server. For more information, see The Destination IP Hash Method, The Source IP Hash Method, The Source IP Destination IP Hash Method, and The Source IP Source Port Hash Method.

To set the load balancing method by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <name> -**lbMethod** <method>

Example

set lb vserver Vserver-LB-1 -lbMethod LeastConnection

Parameters for specifying a load balancing method

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

lbMethod

Load balancing method used by the virtual server. Valid values: ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, URLHASH, DOMAINHASH, DESTINATIONIPHASH, SOURCEIPHASH, SRCIPDESTIPHASH, LEASTBANDWIDTH, LEASTPACKETS, TOKEN, SRCIPDESTIPHASH, CUSTOMLOAD. Default: LEASTCONNECTION.

To set the load balancing method by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure an LB method (for example, Vserver-LB-1), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Method and Persistence** tab.
4. From the drop-down menu under **LB Method**, select a method, (for example, **Least Response Time**).
5. Click **OK**.

Persistence and Persistent Connections

Unless you configure persistence, a load balancing stateless protocol, such as HTTP, disrupts the maintenance of state information about client connections. Different transmissions from the same client might be directed to different servers even though all of the transmissions are part of the same session. You must configure persistence on a load balancing virtual server that handles certain types of Web applications, such as shopping cart applications.

Before you can configure persistence, you need to understand the different types of persistence, how they are used, and what the implications of each type is. You then need to configure the NetScaler appliance to provide persistent connections for those Web sites and Web applications that require them.

You can also configure backup persistence, which takes effect in the event that the primary type of persistence configured for a load balancing virtual server fails. You can configure persistence groups, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

For information about persistence with RADIUS load balancing, see Configuring RADIUS Load Balancing with Persistence.

About Persistence

You can choose from among any of several types of persistence for a given load balancing virtual server, which then routes to the same service all connections from the same user to your shopping cart application, Web-based email, or other network application. The persistence session remains in effect for a period of time, which you specify.

If a server participating in a persistence session goes DOWN, the load balancing virtual server uses the configured load balancing method to select a new service, and establishes a new persistence session with the server represented by that service. If the server goes OUT OF SERVICE, it continues to process existing persistence sessions, but the virtual server does not direct any new traffic to it. After the shutdown period elapses, the virtual server ceases to direct connections from existing clients to the service, closes existing connections, and redirects those clients to new services if necessary.

Depending on the persistence type you configure, the NetScaler appliance might examine the source IPs, destination IPs, SSL session IDs, Host or URL headers, or some combination of these things to place each connection in the proper persistence session. It might also base persistence on a cookie issued by the Web server, on an arbitrarily assigned token, or on a logical rule. Almost anything that allows the appliance to match connections with the proper persistence session and be used as the basis for persistence.

The following table summarizes the persistence types available on the NetScaler appliance.

Table 1. Types of Persistence

Persistence Type	Description
Source IP	SOURCEIP. Connections from the same client IP address are parts of the same persistence session.
HTTP Cookie	COOKIEINSERT. Connections that have the same HTTP Cookie header are parts of the same persistence session.
SSL Session ID	SSLSESSION. Connections that have the same SSL Session ID are parts of the same persistence session.
URL Passive	URLPASSIVE. Connections to the same URL are treated as parts of the same persistence session.
Custom Server ID	CUSTOMSERVERID. Connections with the same HTTP HOST header are treated as parts of the same persistence session.
Destination IP	DESTIP. Connections to the same destination IP are treated as parts of the same persistence session.

Source and Destination IPs	SRCIPDESTIP. Connections that are both from the same source IP and to the same destination IP are treated as parts of the same persistence session.
SIP Call ID	CALLID. Connections that have the same call ID in the SIP header are treated as parts of the same persistence session.
RTSP Session ID	RTSPSID. Connections that have the same RTSP Session ID are treated as parts of the same persistence session.
User-Defined Rule	RULE. Connections that match a user-defined rule are treated as parts of the same persistence session.

Depending on the type of persistence that you have configured, the virtual server can support either 250,000 simultaneous persistent connections or any number of persistent connections up to the limits imposed by the amount of RAM on your NetScaler appliance. The following table shows which types of persistence fall into each category.

Table 2. Persistence Types and Numbers of Simultaneous Connections Supported

Persistence Type	Number of Simultaneous Persistent Connections Supported
Source IP, SSL Session ID, Rule, destination IP, source IP/destination IP, SIP Call ID, RTSP Session ID	250 K
Cookie, URL Server ID, Custom Server ID	Memory limit. In case of CookieInsert, if timeout is not 0, the number of connections is limited by memory.

Some types of persistence are specific to particular types of virtual server. The following table lists each type of persistence and indicates which types of persistence are supported on which types of virtual server.

Table 3. Relationship of Persistence Type to Virtual Server Type

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEIP	YES	YES	YES	YES	YES	YES	NO	NO
COOKIEINSERT	YES	YES	NO	NO	NO	NO	NO	NO
SSLSESSION	NO	YES	NO	NO	YES	YES	NO	NO
URLPASSIVE	YES	YES	NO	NO	NO	NO	NO	NO
CUSTOMSERVERID	YES	YES	NO	NO	NO	NO	NO	NO

RULE	YES	YES	YES **Note:** Load balancing virtual servers of type TCP support rule based persistence only on NetScaler 9.3.e.	NO	NO	YES **Note:** Load balancing virtual servers of type SSL_TCP support rule based persistence only on NetScaler 9.3.e.	NO	NO
SRCIPDESTIP	YES	YES	YES	YES	YES	YES	NO	NO
DESTIP	YES	YES	YES	YES	YES	YES	NO	NO
CALLID	NO	NO	NO	NO	NO	NO	NO	YES
RTSPID	NO	NO	NO	NO	NO	NO	YES	NO

Persistence Based on Source IP Address

When source IP persistence is configured, the load balancing virtual server uses the configured load balancing method to select a service for the initial request, and then uses the source IP address (client IP address) to identify subsequent requests from that client and send them to the same service. You can set a time-out value, which specifies the maximum inactivity period for the session. When the time-out value expires, the session is discarded, and the configured load balancing algorithm is used to select a new server.

Caution: In some circumstances, using persistence based on source IP address can overload your servers. All requests to a single Web site or application are routed through the single gateway to the NetScaler appliance, even though they are then redirected to multiple locations. In multiple proxy environments, client requests frequently have different source IP addresses even when they are sent from the same client, resulting in rapid multiplication of persistence sessions where a single session should be created. This issue is called the “Mega Proxy problem.” You can use HTTP cookie-based persistence instead of Source IP-based persistence to prevent this from happening.

To configure persistence based on Source IP Address, see Configuring Persistence Types That Do Not Require a Rule.

Note: If all incoming traffic comes from behind a Network Address Translation (NAT) device or proxy, the traffic appears to the NetScaler appliance to come from a single source IP address. This prevents Source IP persistence from functioning properly. Where this is the case, you must select a different persistence type.

Persistence Based on HTTP Cookies

When HTTP cookie persistence is configured, the NetScaler appliance sets a cookie in the HTTP headers of the initial client request. The cookie contains the IP address and port of the service selected by the load balancing algorithm. As with any HTTP connection, the client then includes that cookie with any subsequent requests.

When the NetScaler appliance detects the cookie, it forwards the request to the service IP and port in the cookie, maintaining persistence for the connection. You can use this type of persistence with virtual servers of type HTTP or HTTPS. This persistence type does not consume any NetScaler resources and therefore can accommodate an unlimited number of persistent clients.

Note: If the client's Web browser is configured to refuse cookies, HTTP cookie-based persistence will not work. It might be advisable to configure a cookie check on the Web site, and warn clients that do not appear to be storing cookies properly that they will need to enable cookies for the Web site if they want to use it.

The format of the cookie that the NetScaler appliance inserts is:

NSC_XXXX=<*ServiceIP* ><*ServicePort*>

where:

- *NSC_XXXX* is the virtual server ID that is derived from the virtual server name.
- *ServiceIP* is an encrypted representation of the service IP address.
- *ServicePort* is an encrypted representation of the service port.

You can set a time-out value for this type of persistence to specify an inactivity period for the session. When the connection has been inactive for the specified period of time, the NetScaler appliance discards the persistence session. Any subsequent connection from the same client results in a new server being selected based on the configured load balancing method, and a new persistence session being established.

Note: If you set the time-out value to 0, the NetScaler appliance does not specify an expiration time, but sets a session cookie that is not saved when the client's browser is shut down.

By default, the NetScaler appliance sets HTTP version 0 cookies for maximum compatibility with client browsers. (Only certain HTTP proxies understand version 1 cookies; most commonly used browsers do not.) You can configure the appliance to set HTTP version 1 cookies, for compliance with RFC2109. For HTTP version 0 cookies, the appliance inserts the cookie expiration date and time as an absolute Coordinated Universal Time (GMT). It calculates this value as the sum of the current GMT time on the appliance and the time-out value. For HTTP version 1 cookies, the appliance inserts a relative expiration time by setting the "Max-Age" attribute of the HTTP cookie. In this case, the client's browser calculates the actual expiration time.

To configure persistence based on a cookie inserted by the appliance, see Configuring Persistence Types That Do Not Require a Rule.

In the HTTP cookie, the appliance by default sets the httponly flag to indicate that the cookie is nonscriptable and should not be revealed to the client application. Therefore, a client-side script cannot access the cookie, and the client is not susceptible to cross-site scripting.

Certain browsers, however, do not support the httponly flag and, therefore, might not return the cookie. As a result, persistence is broken. For browsers that do not support the flag, you can omit the httponly flag in the persistence cookie.

To change the httponly flag setting by using the NetScaler command line

At the NetScaler command prompt, type:

set lb parameter **-httpOnly** (ENABLED|DISABLED)

Example

```
>  set lb parameter -httponly disabled
   Done
> show lb parameter
  Global LB parameters:
       Persistence Cookie HttpOnly Flag: DISABLED
       Use port for hash LB: YES
 Done
```

Parameter for customizing the httponly flag

httpOnly

Flag the persistence cookie as nonscriptable so that it is not revealed to the client application. Possible values: ENABLED, DISABLED. Default: ENABLED.

To change the httponly flag setting by using the NetScaler configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the **Settings** group, click **Configure Load Balancing Parameters**.
3. To not set the httponly flag in the persistence cookie, clear the **Persistence Cookie HTTPOnly Flag** check box.
4. Click **OK**.
5. Open the **Configure Load Balancing Parameters** dialog box and verify the setting you just configured.

Persistence Based on SSL Session IDs

When SSL Session ID persistence is configured, the NetScaler appliance uses the SSL Session ID, which is part of the SSL handshake process, to create a persistence session before the initial request is directed to a service. The load balancing virtual server directs subsequent requests that have the same SSL session ID to the same service. This type of persistence is used for SSL bridge services.

Note:

There are two issues that users should consider before choosing this type of persistence. First, the NetScaler appliance does not encrypt or decrypt data when it forwards requests to services in an SSL bridge configuration, because it must maintain the data structures to keep track of the sessions. This type of persistence therefore consumes resources on the NetScaler appliance, which limits the number of concurrent persistence sessions that it can support. If you expect to support a very large number of concurrent persistence sessions, you might want to choose another type of persistence.

Second, if the client and the load-balanced server should renegotiate the session ID during their transactions, persistence is not maintained, and a new persistence session is created when the client's next request is received. This may result in the client's activity on the Web site being interrupted and the client being required to reauthenticate or restart the session. It may also result in large numbers abandoned sessions if the timeout is set to too large a value.

To configure persistence based on SSL session ID, see Configuring Persistence Types That Do Not Require a Rule.

Custom Server ID Persistence

In the Custom Server ID persistence method, the Server ID specified in the client request is used to maintain persistence. The NetScaler appliance checks the URL of the client request and connects to the physical server that has the specified server ID. The service provider should make sure that the users are aware of the server IDs to be provided in their requests for specific services.

For example, if your site provides different types of data, such as images, text, and multimedia, from different servers, you can assign each server a server ID. On the NetScaler appliance, you specify those server IDs for the corresponding services, and you configure custom server ID persistence on the corresponding load balancing virtual server. When sending a request, the client inserts into the URL the server ID indicating the required type of data.

To configure custom server ID persistence:

- In your load balancing setup, assign a server ID to each service for which you want to use the user-defined server ID to maintain persistence.
- Specify rules, in the default-syntax expression language, to examine the URL queries for the server ID and forward traffic to the corresponding server.
- Configure custom server ID persistence.

Note: The persistence time-out value does not affect the Custom Server ID persistence. There is no limit on the maximum number of persistent clients because this persistence type does not store any client information.

Example

In a load balancing setup with two services, assign server ID 2345-photo-56789 to Service-1, and server ID 2345-drawing-abb123 to Service-2. Bind these services to a virtual server named Web11. On virtual server Web11, enable Custom Server ID persistence and create the following expression to make the NetScaler to examine requests for URL queries that contain the string "sid=".

URLQUERY contains sid=

When a client sends a request with the following URL to the IP address of Web11, the NetScaler directs the request to Service-2 and creates a persistence session.

http://www.example.com/index.asp?&sid=2345

For more information about default-syntax policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.For more information about persistence methods, see About Persistence.

Parameters for configuring custom server ID persistence

ServiceName

The name of the service that you are configuring.

vServerName

The name of the virtual server

persistenceType

Method of persistence. Select CUSTOMSERVERID.

rule

Rule to examine the URL for server ID.

To configure custom server ID persistence by using the configuration utility

1. Assign a server ID to each of the services for which you want to configure custom server ID persistence.
   a. In the navigation pane, expand **Load Balancing**, and then click **Services**.
   b. In the details pane, select the service for which you want to specify a server ID, and then click **Open**.
   c. In the **Configure Service** dialog box, click the **Advanced** tab.
   d. Scroll down, and under **Others**, in the **Server ID** box, type an ID for the server.
   e. Click **OK**.
2. Configure custom server ID persistence on the virtual server to which the services are bound.
   a. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
   b. In the details pane, select the virtual server for which you want to specify persistence, and then click **Open**.
   c. On the **Method and Persistence** tab, in the **Persistence** group, select **CUSTOMSERVERID**.
   d. In the **Rule** box, type an expression, or click **Configure**, and use the options available in the **Create Expression** dialog box, to create the expression.
   e. Click **OK**.

Persistence Based on Destination IP Addresses

With destination IP address-based persistence, when the NetScaler appliance receives a request from a new client, it creates a persistence session based on the IP address of the service selected by the virtual server (the destination IP address). Subsequently, it directs requests to the same destination IP to the same service. This type of persistence is used with link load balancing. For more information about link load balancing, see Link Load Balancing.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on the destination IP address, see Configuring Persistence Types That Do Not Require a Rule.

Persistence Based on Source and Destination IP Addresses

With source and destination IP address-based persistence, when the NetScaler appliance receives a request, it creates a persistence session based on both the IP address of the client (the source IP address) and the IP address of the service selected by the virtual server (the destination IP address). Subsequently, it directs requests from the same source IP and to the same destination IP to the same service.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on both source and destination IP addresses, see Configuring Persistence Types That Do Not Require a Rule.

Persistence Based on SIP Call ID

With SIP Call ID persistence, the NetScaler appliance chooses a service based on the call ID in the SIP header. This enables it to direct packets for a particular SIP session to the same service and, therefore, to the same load balanced server. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure persistence based on SIP Call ID, see Configuring Persistence Types That Do Not Require a Rule.

Persistence Based on RTSP Session IDs

With RTSP Session ID persistence, when the NetScaler appliance receives a request from a new client, it creates a new persistence session based on the Real-Time Streaming Protocol (RTSP) session ID in the RTSP packet header, and then directs the request to the RTSP service selected by the configured load balancing method. It directs subsequent requests that contain the same session ID to the same service. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

Note: RTSP Session ID persistence is configured by default on RTSP virtual servers, and you cannot modify that setting.

Sometimes different RTSP servers issue the same session IDs. When this happens, unique sessions cannot be created between the client and the RTSP server by using only the RTSP session ID. If you have multiple RTSP servers that may issue the same session IDs, you can configure the appliance to append the server IP address and port to the session ID, creating a unique token that can be used to establish persistence. This is called session ID mapping.

To configure persistence based on RTSP Session IDs, see Configuring Persistence Types That Do Not Require a Rule.

Important: If you need to use session ID mapping, you must set the following parameter when configuring each service within the load balancing setup. Also, make sure that no non-persistent connections are routed through the RTSP virtual server.

Parameter to Set When Configuring Services

session

Map the RTSP session ID by appending the IP address and port of the server to the session ID, guaranteeing that all session IDs are unique within the load balancing setup.

Note: When Session ID Mapping is enabled, the NetScaler appliance rejects any packet that does not contain a mapped ID. Possible values: ON and OFF. Default: OFF.

Note: If a client sends multiple SETUP requests on a single TCP connection, the NetScaler appliance sends those SETUP requests to the same service, because it makes a load balancing decision for every TCP connection. When this occurs, the appliance does not forward the SETUP requests to different servers based on the RTSP session ID.

Not Configuring URL Passive Persistence

With URL Passive persistence, when the NetScaler appliance receives a request from a new client, it extracts the server ID from the server response to the new request. It then embeds the server ID in the query portion of the request URL and creates a persistence session based on that server ID. (The server ID is the IP address and port of the service, expressed as a single hexadecimal number.) Finally, the appliance redirects the modified request to the service selected by the configured load balancing method. Subsequently, when the appliance receives additional requests from the same client, it recognises the server ID and directs those requests to the same service to which it sent the original request.

URL passive persistence requires configuring either a classic or an advanced expression that specifies the query element that contains the server ID. For more information about classic and advanced policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: If your services are assigned IPv6 addresses, you must use an advanced expression to configure URL passive persistence.

The following expression configures the NetScaler to examine requests for URL queries that contain the string “sid=”, extract the server ID, convert it from a hexadecimal string to an IP and port #, and create a persistence session based upon that server ID.

URLQUERY contains sid=

If URL passive persistence is enabled and this expression is configured, a request with the following URL and server ID string is directed to 10.102.29.10:80.

http://www.example.com/index.asp?&sid=c0a864100050

The persistence time-out value does not affect this persistence type; persistence is maintained as long as the server ID can be extracted from client requests. This persistence type does not consume any NetScaler resources, so it can accommodate an unlimited number of persistent clients.

To configure persistence based on the server ID in the URL, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule. You set the persistence type to **URLPASSIVE**. You then perform the procedures provided below.

To configure URL passive persistence by using the command line

At the NetScaler command prompt, type:

set lb vserver <vserverName> [-**rule** <expression>]

Example

set lb vserver LB-VServer-1 –rule URLQUERY contains sid=

Parameters for Rule-Based Persistence

rule

Value used to set the URLPASSIVE persistence type. The value can be an existing rule name, or it can be a classic or advanced expression. The default value is none. The maximum length is 14999.

The rule evaluates either requests that are directed to the load balanced servers or responses from the servers.

To configure URL passive persistence by using the configuration utility

1. If you have not already done so, in the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, click the **Configure** button next to the **Rule** field.
3. In the dialog box that appears, select **Classic Syntax** or **Advanced Syntax**.
4. Select or create the rule that you want to use. For more information, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.
5. Click **OK**.

Configuring Persistence Based on User-Defined Rules

When rule based persistence is configured, the NetScaler appliance creates a persistence session based on the contents of the matched rule before directing the request to the service selected by the configured load balancing method. Subsequently, it directs all requests that match the rule to the same service. You can configure rule based persistence for services of type HTTP, SSL, RADIUS, ANY, TCP, and SSL_TCP.

Note: Load balancing virtual servers of service type TCP and SSL_TCP support rule based persistence only on NetScaler 9.3.e.

Rule based persistence requires a classic or default syntax expression. You can use a classic expression to evaluate request headers, or you can use a default syntax expression to evaluate request headers, Web form data in a request, response headers, or response bodies. For example, you could use a classic expression to configure persistence based on the contents of the HTTP Host header. You could also use a default syntax expression to configure persistence based on application session information in a response cookie or custom header. For more information on creating and using classic and default syntax expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX130086.

The expressions that you can configure depends on the type of service for which you are configuring rule based persistence. For example, certain RADIUS-specific expressions are not allowed for protocols other than RADIUS, and TCP-option based expressions are not allowed for service types other than the ANY type. For TCP and SSL_TCP service types, you can use expressions that evaluate TCP/IP protocol data, Layer 2 data, TCP options, and TCP payloads.

Note: For a use case that involves configuring rule based persistence on the basis of Financial Information eXchange ("FIX") Protocol data transmitted over TCP, see Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream.

Rule based persistence can be used for maintaining persistence with entities such as Branch Repeater appliances, Branch Repeater plug-ins, cache servers, and application servers.

Note: On an ANY virtual server, you cannot configure rule-based persistence for the responses.

To configure persistence based on a user-defined rule, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule, and set the persistence type to RULE. You then perform the procedures provided below. You can configure rule based persistence by using the configuration utility or the NetScaler command line.

To configure persistence based on user-defined rules by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vserverName> [-**rule** <expression>][-**resRule** <expression>]

Example

set lb vserver vsvr_name –rule http.req.header("cookie").value(0).typecast_nvlist_t('=',';').value("server")

set lb vserver vsvr_name –resrule http.res.header("set-cookie").value(0).typecast_nvlist_t('=',';').value("serve

Parameters for Rule-Based Persistence

rule

Value used to set the RULE persistence type. The value can be an existing rule name, or it can be a classic or default syntax expression. You can specify this parameter for services of type HTTP, SSL, RADIUS, ANY, TCP, and SSL_TCP. You can specify an expression that evaluates requests from clients.

Maximum length: 1499 characters. Default: NONE.

resRule

Value used to set the RULE persistence type. The response rule evaluates responses from the load balanced servers. You can configure this parameter for services of type . The expression must be a default syntax expression that evaluates responses from the load balanced servers.

Maximum length: 1499 characters. Default: NONE.

To configure persistence based on user-defined rules by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, choose the type of rule you want to configure.
   - If you want to base the rule on the request, click the **Configure** button next to the **Rule** field.
   - If you want to base the rule on the response, click the **Configure** button next to the **Response Rule** field.
4. In the dialog box that appears, select **Switch to Classic Syntax** or **Switch to Advanced Syntax**.
5. Select or create the rule that you want to use. Some examples of rules that you might find useful are provided below. For more information, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX130086.
6. Click **OK**.

Example: Classic Expression for a Request Payload

The following classic expression creates a persistence session based on the presence of a User-Agent HTTP header that contains the string, “MyBrowser”, and directs any subsequent client requests that contain this header and string to the same server that was selected for the initial request.

```
http header User-Agent contains MyBrowser
```

Example: Default syntax Expression for a Request Header

The following default syntax expression does exactly the same thing as the previous classic expression.

```
HTTP.REQ.HEADER("User-Agent").CONTAINS ("MyBrowser")
```

Example: Default syntax Expression for a Response Cookie

The following expression examines responses for “server” cookies, and then directs any requests that contain that cookie to the same server that was selected for the initial request.

```
HTTP.RES.HEADER("SET-COOKIE").VALUE(0).TYPECAST_NVLIST_T('=',';').VALUE("server")
```

Configuring Persistence Types That Do Not Require a Rule

To configure persistence, you must first set up a load balancing virtual server, as described in Setting Up Basic Load Balancing. You then configure persistence on the virtual server.

To configure persistence on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure persistence and verify the configuration:

- set lb vserver <name> -**PersistenceType** <type> [-**timeout** <integer>]
- show lb vserver

Example

set lb vserver Vserver-LB-1 -persistenceType SOURCEIP

show lb vserver

Note: For IP-based persistence, you can also set the **persistMask** parameter.

Parameters for configuring persistence

PersistenceType

Persistence type supported on the virtual server. Valid Values: SOURCEIP, COOKIEINSERT, SSLSESSION, RULE, URLPASSIVE, CUSTOMSERVERID, DESTIP, SRCIPDESTIP, CALLID, RTSPID, and NONE. Default: NONE.

persistMask

Defines which IP range is used to determine whether a connection is part of an existing persistence session or not. This parameter is used only if the persistence type is IP-based. The default value, 255.255.255.255, specifies that only connections from the same IP are part of an existing session. Valid values include the full range of available subnet masks.

Note: Setting this parameter to 0 has the same effect as setting it to 255.255.255.255.

timeout

The period, in minutes, for which a persistence session remains in effect. Maximum value: 1440 minutes. Default: 2 minutes.

To configure persistence on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, in the **Persistence** list, select the persistence type you want to use (for example, **SOURCEIP**).
4. In the **Time-out** and **Netmask** text boxes type the time-out and subnet mask values (for example, **2** and **255.255.255.255**).
5. Click **OK**.

Configuring Backup Persistence

The NetScaler appliance uses backup persistence to choose a new type of persistence when the primary persistence type fails. For example, if the primary persistence type is set to Cookie Insert, and backup persistence is set to Source IP, the NetScaler appliance uses Source IP-based persistence when the cookie is missing from the HTTP header or when the client browser does not support cookies.

You can set a time-out value for backup persistence only when the primary persistence type is HTTP Cookie-based or RTSP session ID-based persistence, and the backup persistence type is Source IP-based.

To set backup persistence for a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vserverName> -**persistenceType** <PersistenceType> -**persistenceBackup** <BackupPersistenceType>

Example

set lb vserver Vserver-LB-1 -persistenceType CookieInsert -persistenceBackup SourceIP

Parameter for configuring backup persistence

persistenceBackup

Backup persistence type for the group. Possible values: SOURCEIP, NONE. Default: NONE.

To set backup persistence for a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure backup persistence (for example, Vserver-LB-1), and then click **Open**.
3. The **Configure Virtual Server (Load Balancing)** dialog box, click the **Method and Persistence** tab.
4. In the **Persistence** list, select the persistence type you want (for example, **COOKIEINSERT**).
5. In the **Time-out** text box, type the time-out value you want (for example, **20**).
6. In the **Backup Persistence** list, select the backup persistence type that you want to configure (for example, **SOURCEIP**).
7. In the **Backup Time-out** and **Netmask** text boxes, type the backup time-out value and netmask (for example, **20** and **255.255.255.255**).
8. Click **OK**.

Configuring Persistence Groups

When you have load-balanced servers that handle several different types of connections (such as Web servers that host multimedia), you can configure a virtual server group to handle these connections. To create a virtual server group, you bind different types of virtual servers, one for each type of connection that your load balanced servers accept, into a single group. You then configure a persistence type for the entire group.

You can configure either source IP-based persistence or HTTP cookie-based persistence for persistence groups. After you set persistence for the entire group, you cannot change it for individual virtual servers in the group. If you configure persistence on a group and then add a new virtual server to the group, the persistence of the new virtual server is changed to match the persistence setting of the group.

When persistence is configured on a group of virtual servers, persistence sessions are created for initial requests, and subsequent requests are directed to the same service as initial request, regardless of the virtual server in the group that receives each client request.

If you configure HTTP cookie-based persistence, the domain attribute of the HTTP cookie is set. This setting causes the client software to add the HTTP cookie into client requests if different virtual servers have different public host names. For more information about CookieInsert persistence type, see Persistence Based on HTTP Cookies.

To create a virtual server persistency group by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb group <vServerGroupName> <vServerName> -**persistenceType** <PersistenceType>

Example

bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType CookieInsert

Parameters for configuring virtual server persistency groups

name

Name of the persistence group. This alphanumeric string is required and cannot be changed after the persistence group is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

persistenceType

Persistence type for the group. Possible values: SOURCEIP, COOKIEINSERT, NONE.

Note: If you specify NONE, the persistence configured for each of the individual virtual servers is applied, and the persistence group does not function.

persistMask

Netmask specified when the persistency type is SOURCEIP.

timeout

Time period for which the persistence is in effect for a specific client. The value ranges from 2 through 1440 minutes. The default value is 2 minutes. The maximum value is 1440 minutes (1 day).

To create a virtual server persistency group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Persistency Groups**.
2. On the **Persistency Groups** pane, click **Add**.
3. In the **Create Persistency Group** dialog box, in the **Group Name** text box type a name for the group (for example, **Vserver-Group-1**).
4. In the **Persistence** list, select a persistence type (for example, **SOURCEIP**).
5. In the **Persistence Mask** and **Time-out** text boxes, type the persistence mask and timeout values (for example, **255.255.255.255** and **2**).
6. Under **Virtual Server List**, in the **Available Virtual Server** list box, select the virtual server that you want to bind to the group (for example, **Vserver-LB-1**), and then click **Add**.
7. Click **Create**, and then click **Close**. The virtual server group you created appears in the **Persistence Groups** pane.

You can also change the backup persistence type, backup persistence time-out, and cookie domain value on an existing persistence group.

To modify a virtual server group by using the NetScaler command line

At the NetScaler command prompt, type:

set lb group <vServerGroupName> -**PersistenceBackup** <BackupPersistenceType> -**persistMask** <SubnetMaskAddress>

Example

set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask 255.255.255.255

Parameters for modifying virtual server persistency groups

PersistenceBackup

Backup persistence type for the group. The valid options for this parameter are: SOURCEIP and NONE

backupPersistenceTimeout

Maximum time, in minutes, for which the backup persistence is in effect for a specific client. Maximum value: 1440. Default: 2.

cookieDomain

Domain attribute of the HTTP cookie.

To modify a virtual server group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Persistence Groups**.
2. In the **Persistence Groups** pane, select the virtual server group that you want to modify (for example, **Vserver-Group-1**), and click **Open**.
3. The **Configure Virtual Server Group** dialog box appears.
4. In the **Backup Persistence** list, select the type of backup persistence (for example, **SOURCEIP**).
5. In the **Persistence Mask** text box, type the subnet mask (for example, **255.255.255.255**).
6. Click **OK**.

Configuring RADIUS Load Balancing with Persistence

Today's complex networking environment often requires coordinating a high-volume, high-capacity load balancing configuration with robust authentication and authorization. Application users may connect to a VPN through mobile access points such as consumer-grade DSL or Cable connections, WiFi, or even dial-up nodes. Those connections usually use dynamic IPs, which can change during the connection.

If you configure RADIUS load balancing on the NetScaler appliance to support persistent client connections to RADIUS authentication servers, the appliance uses the user logon or the specified RADIUS attribute instead of the client IP as the session ID, directing all connections and records associated with that user session to the same RADIUS server. Users are therefore able to log on to your VPN from mobile access locations without experiencing disconnections when the client IP or WiFi access point changes.

To configure RADIUS load balancing with persistence, you must first configure RADIUS authentication for your VPN. For information and instructions, see the Authentication, Authorization, Auditing (AAA) chapter in the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674. You must also choose either the Load Balancing or Content Switching feature as the basis for your configuration, and make sure that the feature you chose is enabled. The configuration process with either feature is almost the same.

Then, you configure either two load balancing, or two content switching, virtual servers, one to handle RADIUS authentication traffic and the other to handle RADIUS accounting traffic. Next, you configure two services, one for each load balancing virtual server, and bind each load balancing virtual server to its service. Finally, you create a load balancing persistency group and set the persistency type to RULE.

Enabling the Load Balancing or Content Switching Feature

To use the Load Balancing or Content Switching feature, you must first ensure that the feature is enabled. If you are configuring a new NetScaler appliance that has not previously been configured, both of these features are already enabled, so you can skip to the next section. If you are configuring a NetScaler appliance with a previous configuration on it, and you are not certain that the feature you will use is enabled, you must do that now.

- For instructions on enabling the load balancing feature, see Enabling Load Balancing.
- For instructions on enabling the content switching feature, see Enabling Content Switching.

Configuring Virtual Servers

After enabling the load balancing or content switching feature, you must next configure two virtual servers to support RADIUS authentication:

- **RADIUS authentication virtual server.** This virtual server and its associated service will handle authentication traffic to your RADIUS server. Authentication traffic consists of connections associated with users logging onto your protected application or virtual private network (VPN).
- **RADIUS accounting virtual server.** This virtual server and its associated service will handle accounting connections to your RADIUS server. Accounting traffic consists of connections that track an authenticated user's activities on your protected application or VPN.

Important: You must create either a pair of load balancing virtual servers or a pair of content switching virtual servers to use in your RADIUS persistence configuration. You cannot mix virtual server types.

To configure a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt type the following commands to create a new load balancing virtual server and verify the configuration:

- add lb vserver <name> RADIUS <IP address> <port> -**lbmethod** TOKEN -**rule** <rule>
- show lb vserver <name>

To configure an existing load balancing virtual server, replace the above add lb virtual server command with the set lb vserver command, which takes the same arguments.

To configure a content switching virtual server by using the NetScaler command line

At the NetScaler command prompt type the following commands to create a new content switching virtual server and verify the configuration:

- add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -**rule** <rule>
- show cs vserver <name>

To configure an existing content switching virtual server, replace the above add cs vserver command with the set cs vserver command, which takes the same arguments.

Example

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812
    -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813
    -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812
    -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813
    -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
```

Parameters for configuring virtual servers

name

A name for your new virtual server, or the name of the existing virtual server you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

protocol

RADIUS

IPAddress

The IP address assigned to your virtual server. This is normally an Internet-routable IP.

Note: If the virtual server uses IPv6, select the IPv6 check box and enter the address in IPv6 format. (For example, 1000:0000:0000:0000:0005:0600:700a:888b.)

port

The port on which your virtual server listens for connections.

lbmethod

TOKEN

rule

Which policy rule to use as the basis for persistence. The two supported rules are:

- CLIENT.UDP.RADIUS.USERNAME. Use the client login name.

- CLIENT.UDP.RADIUS.ATTR_TYPE(INT). Use the specified RADIUS attribute type. For INT, substitute the integer assigned to that attribute type as specified in RFC4014.

To configure a load balancing or content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing** or **Content Switching**, and then click **Virtual Servers**.

   Note: Except for the GUI location where you create or configure the virtual server, the process is the same.

2. In the details pane, do one of the following:

   - To create a new virtual server, click **Add**.

   - To modify an existing virtual server, select the virtual server, and then click **Open**.

3. In the **Create Virtual Server (Load balancing)** or **Configure Virtual Server (Content Switching)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring virtual servers" as shown:

   - **Name***—name

   - **Protocol***—protocol

   - **IP address***—IPAddress

   - **Port***—port

   * A required parameter

4. In the **Method and Persistence** tab, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring virtual servers" as shown:

   - **Method***—method

   - **Rule***—rule

   * A required parameter

5. Click **Close**. The virtual server that you created now appears in the **Virtual Servers** pane.

Configuring Services

After configuring your virtual servers, you must next configure two services, one for each of the virtual servers that you created. For instructions, see Configuring Services. You should set the service parameters as described in "Parameters for configuring services."

Note: Once configured, these services are in the DISABLED state until the NetScaler appliance can connect to your RADIUS server's authentication and accounting IPs and monitor their status.

Parameters for configuring services

name

A name for your new service, or the name of the existing service you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

IPAddress

The IP address used to connect to the RADIUS server for authentication or accounting, as appropriate, in either IPv4 or IPv6 format. When you provide the IP address of the service, the NetScaler appliance automatically creates a server object with this IP address as its name.

serviceType

The service type, always RADIUS when configuring RADIUS load balancing with persistence.

port

The port on which your service listens for connections.

Binding Virtual Servers to Services

After configuring your services, you must next bind each of the virtual servers that you created to the appropriate service. For instructions, see Binding Services to the Virtual Server.

Configuring a Persistency Group for Radius

After binding your load balancing virtual servers to the corresponding services, you must set up your RADIUS load balancing configuration to support persistence. To do so, you configure a load balancing persistency group that contains your RADIUS load balancing virtual servers and services, and configure that load balancing persistency group to use rule-based persistence. For instructions, see Configuring Persistence Groups. You should set the parameters as described in "Parameters for configuring RADIUS load balancing persistency groups."

Parameters for configuring RADIUS load balancing persistency groups

name

The name of the load balancing persistency group that you are setting or binding. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

vservername

The name of the load balancing virtual server that you are binding to the load balancing persistency group. The name can have the same length and characteristics as the name described above.

newname

The new name of the load balancing persistency group that you are renaming. The name can have the same length and characteristics as the name described above.

rule

Which policy rule to use as the basis for persistence. The two supported rules are:

- CLIENT.UDP.RADIUS.USERNAME. Use the client login name.
- CLIENT.UDP.RADIUS.ATTR_TYPE(INT). Use the specified RADIUS attribute type. For INT, substitute the integer assigned to that attribute type as specified in RFC4014

Your RADIUS load balancing configuration is now complete.

Viewing Persistence Sessions

You can view the different persistence sessions that are in effect globally or for a particular virtual server.

To view a persistence session by using the NetScaler command line

At the NetScaler command line, to view all persistence sessions type:

show lb persistentSessions [<vServer>]

Example

show lb persistentSessions myVserver

Parameters for viewing a persistence session

vServer

Name of the virtual server on which the persistence sessions are running.

To view persistence sessions by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the details pane, under **Monitor Sessions**, click **Virtual Server persistence sessions**.

Clearing Persistence Sessions

You might need to clear persistence sessions from the NetScaler if sessions fail to time out.

To clear a persistence session by using the NetScaler command line

At the NetScaler command prompt, type:

clear lb persistentSessions [<vServer>]

Example

clear lb persistentSessions myLBVserver

Parameter for clearing a persistence session

vServer

The name of the LB vserver whose persistence sessions are to be flushed. If not specified, all persistence sessions will be flushed . Maximum Length: 127

To clear persistence sessions by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the details pane, under **Monitor Sessions**, click **Clear persistence sessions**.
3. In the **Clear Persistence Sessions** dialog box, in **Virtual Servers**, select the virtual server whose persistence sessions you want to clear.
4. Click **OK**.

Customizing the Hash Algorithm for Persistence across Virtual Servers

The NetScaler appliance uses hash-based algorithms for maintaining persistence across virtual servers. By default, the hash-based load balancing method uses a hash value of the IP address and port number of the service. If a service is made available at different ports on the same server, the algorithm generates different hash values. Therefore, different load balancing virtual servers might send requests for the same application to different services, breaking the pseudo-persistence.

As an alternative to using the port number to generate the hash value, you can specify a unique hash identifier for each service. For a service, the same hash identifier value must be specified on all the virtual servers. If a physical server serves more than one type of application, each application type should have a unique hash identifier.

The algorithm for computing the hash value for a service works as follows:

- By default, a global setting specifies the use of port number in a hash calculation.
- If you configure a hash identifier for a service, it is used, and the port number is not, regardless of the global setting.
- If you do not configure a hash identifier, but change the default value of the global setting so that it does not specify use of the port number, the hash value is based only on the IP address of the service.
- If you do not configure a hash identifier or change the default value of the global setting to use the port number, the hash value is based on the IP address and the port number of the service.

You can also specify hash identifiers when using the NetScaler command line to bind services to a service group. In the configuration utility, you can open a service group and add hash identifiers on the Members tab.

To change the use-port-number global setting by using the NetScaler command line

At the NetScaler command prompt, type:

set lb parameter **-usePortForHashLb** (YES | NO)

Example

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
Global LB parameters:
        Persistence Cookie HttpOnly Flag: DISABLED
        Use port for hash LB: NO
 Done
```

To change the use-port-number global setting by using the NetScaler configuration utility

- In the navigation pane, click **Load Balancing**.
- In the **Settings** group, click **Configure Load Balancing Parameters**.
- To not use the port number to generate the hash value, clear the **Use Port for Hash Based LB Methods** check box.
- Click **OK**.
- Open the **Configure Load Balancing Parameters** dialog box and verify the setting you just configured.

To create a new service and specify a hash identifier for a service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the hash ID and verify the setting:

add service **< name >** (< ip > |< serverName >) < serviceType > < port > -hashId < positive_integer >

show service **<name>**

Example

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
        flbkng (10.101.10.1:80) - HTTP
        State: DOWN
        Last state change was at Thu Nov  4 10:14:52 2010
        Time since last state change: 0 days, 00:00:15.990
        Server Name: 10.101.10.1
```

```
        Server ID : 0   Monitor Threshold : 0

        Down state flush: ENABLED
        Hash Id: 12345

1)      Monitor Name: tcp-default
              State: DOWN     Weight: 1

 Done
```

To specify a hash identifier for an existing service by using the NetScaler command line

Type the set service command, the name of the service, and **-hashID** followed by the ID value.

To specify a hash identifier while adding a service group member

To specify a hash identifier for each member to be added to the group and verify the setting, at the NetScaler command prompt, type the following commands (Be sure to specify a unique hashID for each member.):

bind servicegroup <serviceGroupName> <memberName> <port> **-hashId** <positive_integer>

show servicegroup <serviceGroupName>

Example

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
        SRV - HTTP
        State: ENABLED  Monitor Threshold : 0
        …

        1)      1.1.1.1:80   State: DOWN    Server Name: 1.1.1.1    Server ID: 123  Weight: 1
        Hash Id: 32211

              Monitor Name: tcp-default      State: DOWN
        …

        2)      2.2.2.2:80   State: DOWN    Server Name: 2.2.2.2    Server ID: 123  Weight: 1
        Hash Id: 12345
```

```
            Monitor Name: tcp-default       State: DOWN
      …
Done
```

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required. The name must not exceed 127 characters, and the leading character must be a number or a letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ip

IP address of the server that is associated with the service, in either IPv4 or IPv6 format.

serverName

Name of the server that is associated with the service. The name must not exceed 127 characters, and the leading character must be a number or a letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serviceType

Protocol supported by the service. Possible values: HTTP, FTP, TCP, UDP, SSl, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, DNS_TCP, ADNS_TCP, RDP, RADIUS.

port

The port number used for the service.

hashId

The hash identifier for the service. Must be unique for each service. Minimum value: 1, Maximum value: 4294967295.

To specify a hash identifier for a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and click **Services**.
2. In the details pane, do one of the following:
   - To create a new service, click **Add**.
   - To modify an existing service, select the service and then click **Open.**
3. In the **Create Service** or **Configure Service** dialog box, specify values for the following parameters, which correspond to the parameters described in "Parameters for adding a service," as shown:
   - **ServiceName***—name
   - **Server***—ip or serverName
   - **Protocol***—serviceType
   - **Port***—port
4. Click the **Advanced** tab and then scroll down in the dialog box.
5. In the **Hash ID** box, type a unique hash ID value.
6. Click **Create**.
7. Open the service and verify the settings you just configured.

To specify a hash identifier for an already configured service group member by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. Select a **service** group, and then click **Open**.
3. On the **Members** tab, under **Configured Members**, in the row of the member for which you want to specify a hash ID value, double-click the space in the Hash ID column.
4. Type a unique hash ID value.
5. Click **OK**.
6. Open the service group and verify the hash IDs of the service group members you just configured.

Configuring the Redirection Mode

The redirection mode configures the method used by a virtual server to determine where to forward incoming traffic. The NetScaler appliance supports the following redirection modes:

- IP-Based forwarding (the default)
- MAC-Based forwarding

You can configure MAC-Based forwarding on networks that use direct server return (DSR) topology, link load balancing, or firewall load balancing. For more information on MAC-Based forwarding, see the "Interfaces" chapter in the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

To configure the redirection mode by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**m** <RedirectionMode>

Example

set lb vserver Vserver-LB-1 -m MAC

Parameter for configuring the redirection mode

m

The load balancing redirection mode. Possible Values: IP, MAC. Default: IP.

If set to IP, the destination IP address of the request is changed to the IP address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server.

If set to MAC, the destination MAC address is changed to the MAC address of the server to which you are redirecting traffic, and the traffic is then forwarded to that server. With this setting, the destination IP address of the traffic is not changed.

To configure the redirection mode by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, under **Redirection Mode**, click either **IP-Based** or **MAC-Based**.
4. Click **OK**.

Configuring per-VLAN Wildcarded Virtual Servers

If you want to configure load balancing for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a wildcarded virtual server that listens to a specific VLAN and verify the configuration:

- add lb vserver <name> <serviceType> IPAddress * Port * -**listenpolicy** <expression> [-**listenpriority** <positive_integer>]
- show vserver [<name>]

Example

add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)" -listenpriority 10

show vserver Vserver-LB-vlan1

Parameters for configuring per-VLAN wildcarded virtual servers

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

IPAddress

IP address of the virtual server. For wildcarded virtual servers bound to VLANs, this is always *

serviceType

Behavior of the service. Select one of the following service types: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP.

port

Port on which the virtual server listens for client connections. The port number must be in the range 0-65535. For wildcarded virtual servers bound to VLANs, the setting is normally *.

listenpolicy

Use this parameter to specify the listen policy for LB Vserver. The string can be either an existing expression name (configured using add policy expression command) or else it can be an in-line expression with a maximum of 1500 characters.

listenpriority

The priority assigned to the listen policy. This can be any positive integer. Priority is evaluated in reverse order; the lower the number, the higher the priority assigned to the listen policy.

rule

The policy rule to use to identify the VLAN that you want this virtual server to listen to. This rule is:

- CLIENT.VLAN.ID.EQ(<integer>)
- For <integer>, substitute the ID number assigned to the VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, do one of the following:
   - To create a new virtual server, click **Add**.
   - To modify an existing virtual server, select the virtual server, and then click **Open**.
3. In the **Create Virtual Server** or **Configure Virtual Server** dialog box, on the **Services** tab, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring per-VLAN wildcarded virtual servers" as shown:
   - **Name***—name (Cannot be changed for a previously configured virtual server)
   - **Protocol***—serviceType
   - **IP address***—IPAddress
   - **Port**—port

   *A required parameter
4. In the **Advanced** tab, expand **Listen Policy**, and then specify values for the following parameters, which correspond to parameters described in "Parameters for configuring per-VLAN wildcarded virtual servers" as shown:
   - **Listen Priority***—listenpriority
   - **Listen Policy Rule***—rule

   *A required parameter
5. Click **Create** or **OK**, depending on whether you are creating a new virtual server or modifying an existing virtual server.
6. Click **Close**. The virtual server that you created now appears in the Virtual Servers page.
7. To remove a virtual server, in the **Virtual Servers** pane select the virtual server, and then click **Remove**.

After you have created this virtual server, you bind it to one or more services as described in Setting Up Basic Load Balancing.

Assigning Weights to Services

In a load balancing configuration, you assign weights to services to indicate the percentage of traffic that should be sent to each service. Services with higher weights can handle more requests; services with lower weights can handle fewer requests. Assigning weights to services allows the NetScaler appliance to determine how much traffic each load balanced server can handle, and therefore more effectively balance load.

Note: If you use a load balancing method that supports weighting of services (for example, the round robin method), you can assign a weight to the service.

The following table describes the load balancing methods that support weighting, and briefly describes the manner in which weighting affects how a service is selected for each one.

Load Balancing Methods	Service Selection with Weights
Round Robin	The virtual server prioritizes the queue of available services such that services with the highest weights come to the front of the queue more frequently than those with the lowest weights and receive proportionately more traffic. For a complete description, see The Round Robin Method.
Least Connection	The virtual server selects the service with the best combination of fewest active transactions and highest weight. For a complete description, see The Least Connection Method.
Least Response Time and Least Response Time Method using Monitors	The virtual server selects the service with the best combination of fewest active transactions and fastest average response time. For a complete description, see The Least Response Time Method.
Least Bandwidth	The virtual server selects the service with the best combination of least traffic and highest bandwidth. For a complete description, see The Least Bandwidth Method.
Least Packets	The virtual server selects the service with the best combination of fewest packets and highest weight. For a complete description, see The Least Packets Method.

Custom Load	The virtual server selects the service with the best combination of lowest load and highest weight. For a complete description, see The Custom Load Method.
Hashing methods and Token method	Weighting is not supported by these load balancing methods.

To configure a virtual server to assign weights to services by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**weight** <Value> <ServiceName>

Example

set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1

Parameter for setting weights

weight

Weight to be assigned to the specified service. The minimum value is 1 and the maximum value is 100.

To configure a virtual server to assign weights to services by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server (for example, **Vserver-LB-1**), and click **Open**.
3. On the **Services** tab, in the **Weights** spin box, type or select the weight to assign to the service (for example, **10**).
4. Click **OK**.

Protecting the Load Balancing Configuration against Failure

When a load balancing virtual server fails, or when the virtual server is unable to handle excessive traffic, the load balancing setup can fail. You can protect your load balancing setup against failure by configuring the NetScaler appliance to redirect excess traffic to an alternate URL, configuring a backup load balancing virtual server, and configuring stateful connection failover.

Redirecting Client Requests to an Alternate URL

In the event that a load balancing virtual server of type HTTP or type HTTPS goes DOWN or is disabled, you can redirect requests to an alternate URL by using an HTTP 302 redirect. The alternate URL can provide information about the status of the server.

You can redirect to a page on the local server or a remote server. You can redirect to a relative URL or an absolute URL. If you configure a redirect to a relative URL consisting of a domain name with no path, the NetScaler appliance appends the path of the incoming URL to the domain. If you use an absolute URL, the HTTP redirect is sent to that URL with no modification.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when both the primary and backup virtual servers are DOWN.

To configure a virtual server to redirect the client request to a URL by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**redirectURL** <URLValue>

Example

set lb vserver Vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance

Parameters for Redirecting Client Requests to an Alternative URL

redirectURL

URL to which traffic is redirected when the load balancing virtual server is unavailable. This URL length must not exceed 127 characters.

Note: The domain specified in the URL must not match the domain specified in the domain name argument of a content switching policy. If the same domain is specified in both arguments, the request is redirected continuously to the same unavailable load balancing virtual server.

To configure a virtual server to redirect the client request to a URL by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure redirect URL (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, in the **Redirect URL** text box, type the **URL** (for example, **http://www.newdomain.com/mysite/maintenance**).
4. Click **OK**.

Configuring a Backup Load Balancing Virtual Server

You can configure the NetScaler appliance to direct requests to a backup virtual server in the event that the primary load balancing virtual server is DOWN or unavailable. The backup virtual server is a proxy and is transparent to the client. The appliance can also send a notification message to the client regarding the site outage.

You can configure a backup load balancing virtual server when you create it, or you can change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating cascading backup virtual servers. The maximum depth of cascading backup virtual servers is 10.

If you have multiple virtual servers that connect to two servers, you have a choice for what happens if the primary virtual server goes DOWN and then comes back up. The default behavior is for the primary virtual server to resume its role as primary. However, you may want to configure the backup virtual server to remain in control in the event that it takes over. For example, you may want to sync updates on the backup virtual server to the primary virtual server and then manually force the original primary server to resume its role. In this case, you can designate the backup virtual server to remain in control in the event that the primary virtual server goes DOWN and then comes back up.

You can configure a redirect URL on the primary load balancing virtual server as a fallback for when both the primary and the backup virtual servers are DOWN or have reached their threshold for handling requests. When services bound to virtual servers are OUT OF SERVICE, the appliance uses the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when the primary and backup virtual servers are down.

To set a backup virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**backupVserver** <BackupVServerName> [-**disablePrimaryOnDown**]

Example

set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -disablePrimaryOnDown

Parameters for configuring a backup load balancing virtual server

backupVserver

Name of the backup virtual server. You can create a virtual server and specify the name, IP address, port, and type as described in “Creating a Virtual Server,” on page 38. You can use the name of the virtual server as a backup virtual server

disablePrimaryOnDown

Configures the backup virtual server to remain in control, after it takes over, until you manually reenable the primary virtual server.

To set a backup virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the backup virtual server (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, in the **Backup Virtual Server** list, select the backup virtual server (for example, **Vserver-LB-2**).
4. If you want the backup virtual server to remain in control until you manually re-enable the primary virtual server even if the primary virtual server comes back up, select the **Disable Primary When Down** check box.
5. Click **OK**.

Diverting Excess Traffic to a Backup Virtual Server

In addition to taking over for a primary virtual server when it becomes unavailable, a backup load balancing virtual server can handle excess traffic when the primary virtual server reaches its limit. To set this up, you configure the spillover option to divert new connections to the backup virtual server when the number of connections to the primary virtual server exceeds the threshold. You can allow the NetScaler appliance to calculate the threshold dynamically, or you can configure the value manually.

When spillover is configured, the appliance regularly compares the number of established TCP connections on the primary virtual server with the threshold value. When the number of connections reaches the threshold, it diverts new connections to the backup virtual server.

You can configure persistence with spillover. When persistence is configured, connections that are diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections on it drops below the threshold. Instead, the primary virtual server accepts new client connections.

If the backup virtual server reaches its own threshold and is unable to accept additional connections, the NetScaler appliance diverts all requests to the redirect URL. If a redirect URL is not configured on the primary virtual server, any requests over the threshold are dropped.

Note: With RTSP virtual servers, the NetScaler appliance uses only data connections for spillover. If the backup RTSP virtual server is not available, the requests are redirected to an RTSP URL and an RTSP redirect message is sent to the client.

To configure a primary virtual server to divert new connections to a backup virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**soMethod** <spillOverType> -**soThreshold** <positiveInteger> -**soPersistence** ENABLED -**soPersistenceTimeout** <positiveInteger>

Example

set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -soPersistence enabled -soPersistenceTi

Parameters diverting excess traffic to a backup virtual server

soMethod

The type of spillover used to divert traffic to the backup virtual server when the primary virtual server reaches the spillover threshold. Possible values:

- CONNECTION. Spillover based on total number of connections to the virtual server.
- DYNAMICCONNECTION. Spillover based on the sum of the maximum client values configured for each service bound to the virtual server.
- BANDWIDTH. Spillover based on traffic rate.
- HEALTH. Spillover occurs if bound and active services and service groups fall below a threshold relative to all bound elements.
- NONE

soThreshold

The maximum number of connections that the virtual server can accept.

- For the CONNECTION (or) DYNAMICCONNECTION spillover type, the Threshold value is the maximum number of connections a virtual server can handle prior to spillover.
- For the BANDWIDTH spillover type, the Threshold value is the amount of incoming and outgoing traffic (in kilobits per second) that a virtual server can handle before spillover occurs. The minimum value is 1, and the maximum value is 4,294,967,294.
- For HEALTH, the threshold is a positive integer from 1 through 99. This integer represents a percentage of the sum of the binding weights of all of the enabled, bound, and active services and service groups relative to the binding weights of all enabled bound services and service groups (active and inactive).

soPersistence

Spillover persistence state. If you enable spillover persistence, the NetScaler maintains source IP-based persistence over the primary and backup virtual servers. Possible values: enabled, disabled. Default: disabled.

soPersistenceTimeout

Time-out for spillover persistence, in minutes. Minimum value: 2. Maximum value: 1440. Default: 2.

To set a primary virtual server to divert new connections to a backup virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure the spillover (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Advanced** tab, in the **Method** list, select the type of spillover, and in **Threshold** text box, type the threshold value (for example, **Connection** and **1000**).
4. Under **Spillover**, select the **Persistence** check box, and in **Persistence Time-out (min)** text box type the time-out (for example, **2**).
5. Click **OK**.

Configuring Connection-Based Spillover

You can use connection-based spillover to configure a maximum threshold for the number of active client connections on a virtual server. When the client connections exceed the configured threshold limit, new client connections are diverted to the backup virtual server.

To configure connection-based spillover, see Diverting Excess Traffic to a Backup Virtual Server.

Note: Global Server Load Balancing (GSLB) virtual servers do not support connection-based spillover.

Configuring Dynamic Spillover

When you configure dynamic spillover, if the number of client connections to the primary load balancing virtual server exceeds the sum of the maximum client values, new connections are diverted to the backup virtual server.

To configure dynamic spillover, you must first enable it on the primary virtual server. Next, you configure each service that is bound to that virtual server with appropriate maximum client values for that service. Different services can have different maximum client values. If the value for maximum client is set to 0, the spillover limit is treated as infinity, and spillover never occurs.

Note: Content-based virtual servers do not support dynamic spillover.

To configure dynamic spillover, see Diverting Excess Traffic to a Backup Virtual Server.

Configuring Bandwidth-Based Spillover

With bandwidth-based spillover, when the bandwidth used by the primary load balancing virtual server exceeds the specified bandwidth threshold value, the NetScaler appliance diverts new connections to the backup virtual server. You can also configure the backup virtual server with a threshold value. When the threshold for the backup virtual server is reached, the appliance diverts new client connections to the next backup virtual server.

To configure bandwidth-based spillover, see Diverting Excess Traffic to a Backup Virtual Server.

Connection Failover

Connection failover helps prevent disruption of access to applications deployed in a distributed environment. In a NetScaler High Availability (HA) setup, *connection failover* (or *connection mirroring-CM*) refers to keeping active an established TCP or UDP connection when a failover occurs. The new primary NetScaler appliance has information about the connections established before the failover and continues to serve those connections. After failover, the client remains connected to the same physical server. The new primary appliance synchronizes the information with the new secondary appliance by using the SSF framework.

You can set up connection failover in either stateless or stateful mode. In the stateless connection failover mode, the HA nodes do not exchange any information about the connections that are failed over. This method has no runtime overhead.

In the stateful connection failover mode, the primary appliance synchronizes the data of the failed-over connections with the new secondary appliance.

How Connection Failover Works on NetScaler Appliances

In stateless connection failover, the new primary appliance tries to re-create the packet flow according to the information contained in the packets it receives.

In stateful failover, to maintain current information about the mirrored connections, the primary appliance sends messages to the secondary appliance. The secondary appliance maintains the data related to the packets but uses it only in the event of a failover. If a failover occurs, the new primary (old secondary) appliance starts using the stored data about the mirrored connections and accepting traffic. During the transition period, the client and server may experience a brief disruption and retransmissions.

Note:

Verify that the primary appliance is able to authorize itself on the secondary appliance. To verify correct configuration of the passwords, use the show rpcnode command from CLI or use the **RPC** option of the **Network** menu from the configuration utility.

A basic HA configuration with connection failover contains the entities shown in the following figure.

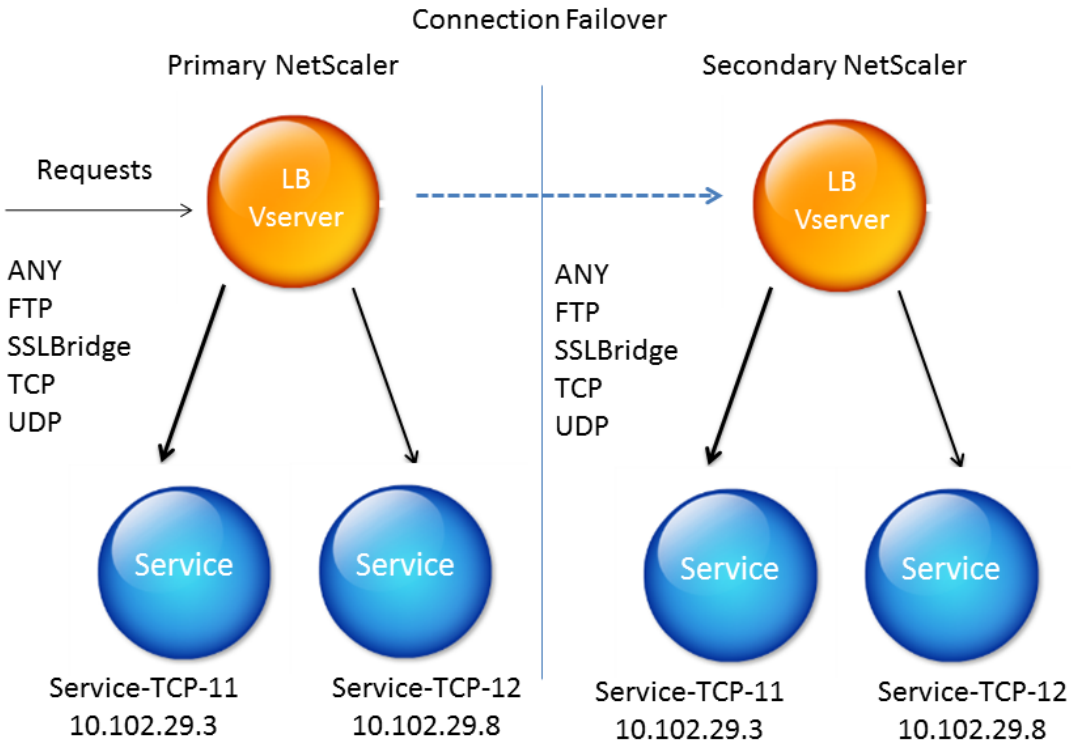


Figure 1. Connection Failover Entity Diagram

Supported Setup

Connection failover can be configured only on load balancing virtual servers. It cannot be configured on content-switching virtual servers.

The following table describes the setup supported for connection failover.

Table 1. Connection Failover - Supported Setup

Setting	Stateless	Stateful
Service Type	ANY.	ANY, UDP, TCP, FTP, SSL_BRIDGE.
Load balancing methods	All methods supported for the service type ANY. However, if Source IP persistence is not set, the SRCIPSRCPORTHASH method must be used.	All methods applicable to the supported service types.

Persistence types	SOURCEIP persistence.	All types applicable to the supported service types are supported.
USIP	Must be ON.	No restriction. It can be ON or OFF.
Service bindings	Service can be bound to only one virtual server.	Service can be bound to one or more virtual servers.

Features Affected by Connection Failover

The following table lists the features affected if connection failover is configured.

Table 2. How Connection Failover Affects NetScaler Features

Feature	Impact of Connection Failover
SYN protection	For any connection, if a failover occurs after the NetScaler issues SYN-ACK but before it receives the final ACK, the connection is not supported by connection failover. The client must reissue the request to establish the connection.
Surge protection	If the failover occurs before a connection with the server is established, the new primary NetScaler tries to establish the connection with the server. It also retransmits all the packets held in the course of surge protection.
Access down	If enabled, the access-down functionality takes precedence over connection failover.
Application Firewall™	The Application Firewall feature is not supported.
INC	Independent network configuration is not supported in the high availability (HA) mode.
TCP buffering	TCP buffering is not compatible with connection mirroring.
Close on response	After failover, the NATPCBs may not be closed on response.
IPv6 virtual servers	Not yet supported.

Configuring Connection Failover

You can configure connection failover on a load balancing virtual server.

To configure connection failover by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**connFailover** <Value>

Example

set lb vserver Vserver-LB-1 -connFailover stateful

Parameters for configuring connection failover

connFailover

State of connection failover on the virtual server. Valid values: STATELESS, STATEFUL, DISABLE. Default: DISABLE.

To configure connection failover by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** pane, select the virtual server for which you want to configure connection failover, and click **Open**.
3. On the **Advanced** tab, in the **Connection Failover** drop-down list, select **Stateful**.
4. Click **OK**.

Disabling Connection Failover

When connection failover is disabled on a virtual server, the resources allocated to the virtual server are freed.

To disable a connection failover by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**connFailover** <Value>

Example

set lb vserver Vserver-LB-1 -connFailover disable

To disable a connection failover by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** pane, select the virtual server for which you want to configure a connection failover and click **Open**.
3. On the **Advanced** tab, in the **Connection Failover** drop-down list box, select **Disable**.
4. Click **OK**.

Flushing the Surge Queue

When a physical server receives a surge of requests, it becomes slow in responding to the currently connected clients and leaves many users dissatisfied and disgruntled. Often, the overloading also causes the clients to receive error pages. To avoid such overloading, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The NetScaler does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the NetScaler looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the NetScaler establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the NetScaler cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the NetScaler cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The NetScaler maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which NetScaler cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.

Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the NetScaler command line

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the NetScaler flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the NetScaler flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the NetScaler flushes the surge queues of all the entities present on the NetScaler.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the NetScaler command prompt, type:

flush ns surgeQ [**-name** <name>] [**-serverName** <serverName> <port>]

Examples

1.
flush ns surgeQ –name SVC1ANZGB –serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and h
2.
flush ns surgeQ
The above command flushes all the surge queues on the NetScaler.

Parameters for flushing a surge queue

name

Name of a virtual server, service or service group

serverName

Name of a service group member

To flush a surge queue by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**.
2. To select an entity, do one of the following:
   - To flush the surge queue of a virtual server, click **Virtual Servers**, and then select the virtual server.
   - To flush the surge queue of a service, click **Services**, and then select the service.
   - To flush the surge queue of all the members in a service group, click **Service Groups**, and then select the service group.
   - To flush the surge queue of a specific member in a service group, click **Service Groups**, and in the action pane, click **Manage Members**. In the **Manage Members of a Service Group** dialog box, select the service group member.

   Note: You can select multiple entities in any window.

   Note: To flush the surge queue of a content switching virtual server, in Steps 1 and 2, expand **Content Switching**, and then select a virtual server.
3. In the action pane, click **Flush Surge Queue**.
4. Click **OK**.

   Note: On the NetScaler, if there are other entities with the same name as you selected, you are alerted that the surge queues of those entities would also be flushed. Take an appropriate action.

Managing a Load Balancing Setup

An existing Load Balancing setup does not require a great deal of work to maintain as long as it is unchanged, but most do not remain unchanged for long. Increasing load requires new load-balanced servers and eventually new NetScaler appliances, which must be configured and added to the existing setup. Old servers wear out and need to be replaced, requiring removal of some servers and addition of others. Upgrades to your networking equipment or changes to topology may also require modifications to your load balancing setup. Therefore, you will need to perform operations on server objects, services, and virtual servers. The Visualizer can display your configuration graphically, and you can perform operations on the entities in the display. You can also take advantage of a number of other features that facilitate management of the traffic through your load balancing setup.

Managing Server Objects

During basic load balancing setup, when you create a service, a server object with the IP address of the service is created, if one does not already exist. If you prefer for your service objects named with domain names rather than IP addresses, you might also have created one or more server objects manually. You can enable, disable, or remove any server object.

When you enable or disable a server object, you enable or disable all services associated with the server object. When you refresh the NetScaler appliance after disabling a server object, the state of its service appears as OUT OF SERVICE. If you specify a wait time when disabling a server object, the server object continues to handle established connections for the specified amount of time, but rejects new connections. If you remove a server object, the service to which it is bound is also deleted.

To enable a server by using the NetScaler command line

At the NetScaler command prompt, type:

enable server <ServerIPAddress>

Example

enable server 10.102.29.5

To enable a server object by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Servers**.
2. In the details pane, select the server that you want to enable (for example, **10.102.29.5**), and then click **Enable**.
3. In the **Enable** dialog box, click **Yes**.

To disable a server object by using the NetScaler command line

At the NetScaler command prompt, type:

disable server <ServerIPAddress> <delay>

Example

disable server 10.102.29.5 30

Wait time parameter

delay

The time, in seconds, after which the server object is marked DOWN.

To disable a server object by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Servers**.
2. In the details pane, select the server that you want to disable (for example, **10.102.29.5**), and then click **Disable**.
3. In the **Wait Time** dialog box, type the wait time after which the server is to be disabled (for example 30).
4. Click **Enter**.

To remove a server object by using the NetScaler command line

At the NetScaler command prompt, type:

rm server <ServerIP>

Example

rm server 10.102.29.5

To remove a server object by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Servers**.
2. In the details pane, select the server that you want to remove (for example, **10.102.29.5**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Managing Services

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time during which the service continues to handle established connections, but rejects new ones, before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the service's state is OUT OF SERVICE.

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the NetScaler configuration.

To enable a service by using the NetScaler command line

At the NetScaler command prompt, type:

enable service <serviceName>

Example

enable service Service-HTTP-1

To enable a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service that you want to enable (for example, **Service-HTTP-1**), and click **Enable**.
3. In the **Enable** dialog box, click **Yes**.

To disable a service by using the NetScaler command line

At the NetScaler command prompt, type:

disable service <serviceName> <DelayInSeconds>

Example

disable service Service-HTTP-1 30

Wait Time Parameter

delay

The time, in seconds, after which the service is marked DOWN.

To disable a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service that you want to disable (for example, **Service-HTTP-1**), and then click **Disable**.
3. In the **Wait Time** dialog box, type the wait time after which the service is to be disabled (for example, **30**).
4. Click **Enter**.

To remove a service by using the NetScaler command line

At the NetScaler command prompt, type:

rm service <ServiceName>

Example

rm service Service-HTTP-1

To remove a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service that you want to remove (for example, **Service-HTTP-1**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Managing a Load Balancing Virtual Server

Virtual servers are enabled by default when you create them. You can disable and reenable virtual servers manually. If you disable a virtual server, the virtual server's state appears as OUT OF SERVICE, and it cannot accept new connections. It continues to serve requests on existing connections.

You remove a virtual server only when you no longer require the virtual server. Before you remove it, you must unbind all services from it.

To enable a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

enable lb vserver <vserverName>

Example

enable lb vserver Vserver-LB-1

To enable a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to enable (for example, **Vserver-LB-1**), and click **Enable**.
3. In the **Enable** dialog box, click **Yes**.

To disable a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

disable vserver <vserverName>

Example

disable lb vserver Vserver-LB-1

To disable a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to disable (for example, **Vserver-LB-1**), and then click **Disable**.
3. In the **Disable** dialog box, click **Yes**.

Note: In the disabled state, a virtual server continues to exist on the network. The NetScaler appliance continues to respond to address resolution protocol (ARP) and Internet control message protocol (ICMP) requests directed to the IP address of the virtual server.

To unbind a service from a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

unbind lb vserver <vserverName> <serviceName>

Example

unbind lb vserver Vserver-LB-1 Service-HTTP-1

To unbind a service from a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server from which you want to unbind a service (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, in the **Services** tab, clear the **Active** check box next to the service that you want to unbind from the virtual server (for example, **Service-HTTP-1**).
4. Click **OK**.

To remove a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

rm lb vserver <vserverName>

Example

rm lb vserver Vserver-LB-1

To remove a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to remove (for example, **Vserver-LB-1**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

The Load Balancing Visualizer

The Load Balancing Visualizer is a tool that you can use to view and modify the load balancing configuration in graphical format. Following is an example of the Visualizer display

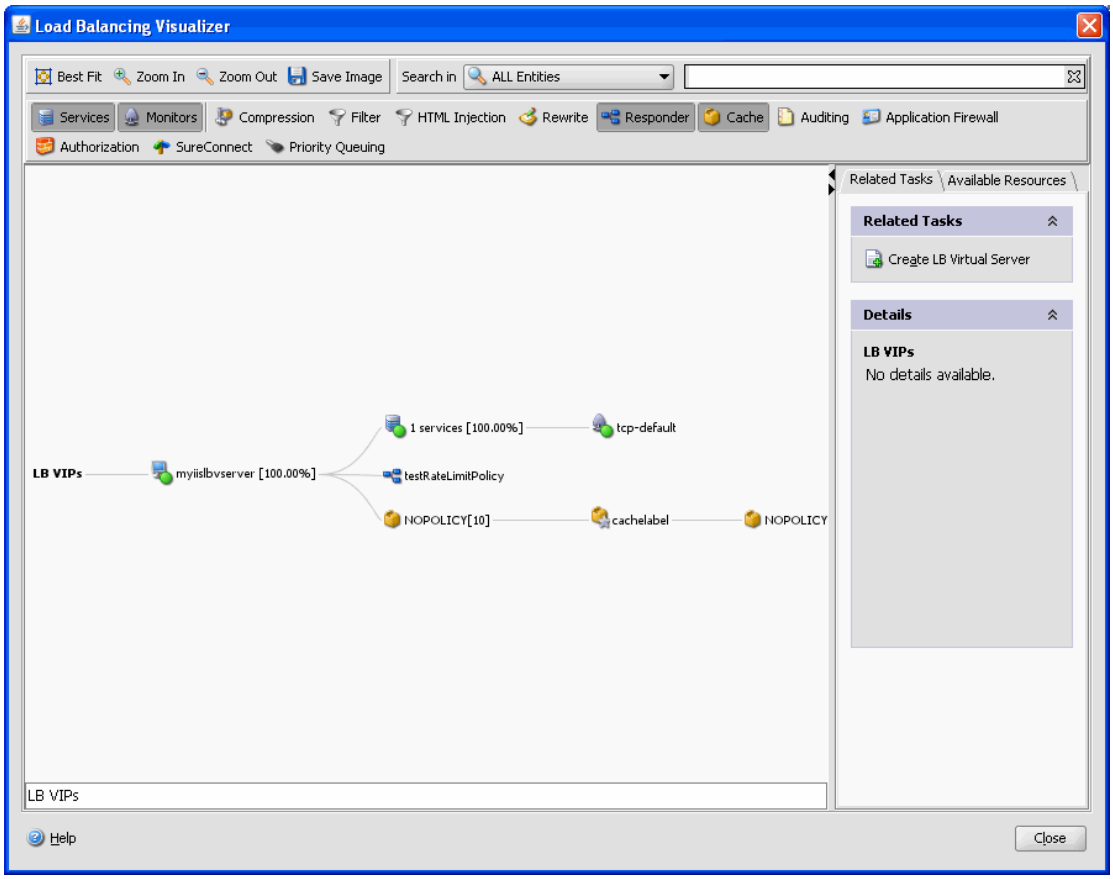


Figure 1. Load Balancing Visualizer Display

You can use the visualizer to view the following:

- The services and service groups that are bound to a virtual server.
- The monitors that are bound to each service.
- The policies that are bound to the virtual server.
- The policy labels, if configured.
- Configuration details of any displayed element.
- Load balancing virtual server statistics.

- Statistical information such as the number of requests received per second by the virtual server and the number of hits per second for rewrite, responder, and cache policies.

- A comparative list of all the parameters whose values either differ or are not defined across service containers.

You can also use the Visualizer to add and bind new objects, modify existing ones, and enable or disable objects. Most configuration elements displayed in the Visualizer appear under the same names as in other parts of the configuration utility. However, unlike the rest of the configuration utility, the Visualizer groups services that have the same configuration details and monitor bindings into an entity called a *service container*.

A service container is set of similar services and service groups that are bound to a single load balancing virtual server. Next to the service container is a number that shows the number of services in the group. The services in the container have the same properties, with the exception of the name, IP address, and port, and their monitor bindings should have the same weight and binding state. When you bind a new service to a virtual server, it is placed into an existing container if its configuration and monitor bindings match those of other services; otherwise, it is placed in its own container.

The service container display can help you troubleshoot your configuration if something is not functioning as you expect. More than one container for a particular virtual server is an indication that something is wrong with the configuration of that virtual server and its services. To correct the problem, you must first identify the container that has the desired configuration. You can do so by using the Service Attributes Diff feature, described below. After you identify the container, you right-click the container and click **Apply Configuration**.

The following procedures provide only basic steps for using the Visualizer. Because the Visualizer duplicates functionality in other areas of the Load Balancing feature, other methods of viewing or configuring all of the settings that can be configured in the Visualizer are provided throughout the Load Balancing documentation.

Note: The Visualizer requires a graphic interface, so it is available only through the configuration utility.

To view load balancing virtual server properties by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, you can adjust the viewable area as follows:

    - Click the **Zoom In** and **Zoom Out** icons to increase or decrease the size of the viewed objects. You can click and drag the viewable area if an item that you want to see disappears from view after zooming in.
    - Click the **Best Fit** icon to optimize the viewing area.
    - Click the **Save Image** icon to save the graph as an image file.
    - Click the image, hold down the mouse button, and drag the image to pan the view.
    - In the **Search in** text field, begin typing the name of the item you are looking for. The item's location is then highlighted. To restrict the search, click the drop-down menu and select the type of element that you want to search for

To view configuration details for services, service groups, and monitors by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, to view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view a summary of bound services, position the cursor over the virtual server icon.
   - To view services in a service container, click the icon for a service group, click the **Related Tasks** tab, click **Show Member Services**, and then click the service group name. To view additional details about the services click **Open**.
   - To view common properties of services in a service group, click the icon for the service group, click the **Related Tasks** tab, and view the **Details** section of the tab.
   - To view a comparative list of the parameters whose values either differ or are not defined across service containers, click the icon for a container, click the **Related Tasks** tab, and then click **Service Attributes Diff**. To view monitor binding details for the services in a container, in the **Service Attributes Diff** dialog box, in the **Group** column for the container, click **Details**.
   - To view the details for a monitor, position the cursor over the icon or click the icon for the monitor. For additional details, click the icon, click the **Related Tasks** tab, and then click **View Monitor**.
   - To view binding details of a monitor, click the connecting line between the monitor and its related service.

To view configuration details for policies and policy labels by using the Visualizer in the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, to view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view policies that are bound to this virtual server, select one or more policy icons in the tool bar at the top of the dialog box. For example, you can select **Compression**, **Filter**, **Rewrite**, and **Responder**. If policy labels are configured, they appear in the main view area.
   - For bound policies that appear in the view pane of the Visualizer, to view a policy's expression and actions, position the cursor over the policy icon. To view binding details, position the cursor over the line that connects the policy to the virtual server. To view these details, click the policy. The details of the policy appear in the details pane.

To view statistical information by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, to view statistical information, you can do the following:
   - To view detailed statistics for the load balancing virtual server, click the icon for the virtual server, click the **Related Tasks** tab, and then click **Statistics**.

     To view the number of requests received per second at a given point in time by the load balancing virtual server and the number of hits per second at a given point in time for rewrite, responder, and cache policies, click **Show Stats**. The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time and has to be refreshed manually. To refresh this information, click **Refresh Stats**.

     Note: The Show Stats option is available only on NetScaler nCore builds.

To save configuration properties for any entity by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click **Related Tasks**.
4. In the **Related Tasks** tab, click **Copy Properties** and then paste the information into a document.

To bind a resource to a load balancing configuration by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure bindings (for example, **Vserver-LB-1**), and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, click the Available Resources tab, select a resource type in the drop-down menu, and do one or more of the following:
   - To bind a new monitor to a service, select **Monitors**, click a particular monitor, and then drag it to the service container icon. Use CONTROL + click to select multiple monitors and drag them to the service.
   - To bind a service or service group, select **Services** or **Service Groups**, respectively, click a particular service or service group, and then drag it to the virtual server icon. To bind multiple services or service groups at one time, press CONTROL + click to select multiple services and drag them over the virtual server.
   - To bind a policy, select one of the policy groups, click a particular policy, and then drag it to a virtual server. To bind multiple policies (classic policies only) at one time, press CONTROL + policies and drag them over the virtual server. For details on classic and advanced policies, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To unbind a resource by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server from which you want to unbind a service, policy, or monitor (for example, **Vserver-LB-1**), and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, on the Visualizer image, click the connecting line between the resources that you want to unbind, and then click **Unbind**. For example, to unbind a monitor, you would click the link between the monitor and its bound service and click **Unbind**.
4. In the **Unbind** dialog box, click **Yes**.

To modify a resource in a load balancing configuration by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure (for example, **Vserver-LB-1**), and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, on the Visualizer image, double-click the resource that you want to modify.

   Note: Alternatively, on the **Available Resources** tab, select the resource type from the drop-down menu, select the particular resource that you want to configure and then click **Open**.
4. In the modify dialog box, enter new settings for the resource.

To add, remove, or disable a resource in a load balancing configuration by using the Visualizer

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure (for example, **Vserver-LB-1**), and then click **Visualizer**.
3. In the **Load Balancing Visualizer** dialog box, right-click the icon for the resource that you want to add, remove, or disable, and then select the corresponding option from the menu.

   Note: Alternatively, on the **Available Resources** tab, click the resource type from the drop-down menu, and then click **Add** to add an entity, or select the particular resource that you want to configure and then click **Open**.

   Note: These options are not available for service groups or policies.

Managing Client Traffic

Managing client connections properly helps to ensure that your applications remain availabile to users even when your NetScaler appliance is experiencing high loads. A number of load balancing features and other features available on the appliance can be integrated into a load balancing setup to process load more efficiently, divert it when necessary, and prioritize the tasks that the appliance must perform:

- **Sessionless load balancing.** You can configure sessionless load balancing virtual servers and perform load balancing without creating sessions in configurations that use DSR or intrusion detection systems (IDS).
- **Integrated caching.** You can redirect HTTP requests to a cache.
- **Priority queuing.** You can direct requests based on priority, by integrating your configuration with the Priority Queuing feature.
- **SureConnect.** You can use load balancing with the SureConnect feature to redirect important requests to a custom Web page, insulating them from delays due to network congestion.
- **Delayed cleanup.** You can configure delayed cleanup of virtual server connections to prevent the cleanup process from using CPU cycles during periods when the NetScaler appliance is experiencing high loads.
- **Rewrite.** You can use the Rewrite feature to modify port and protocol when performing HTTP redirection, or insert the virtual server IP address and port into a custom Request header.
- **RTSP NAT.**
- **Rate-based monitoring.** You can enable rate-based monitoring to divert excess traffic.
- **Layer 2 Parameters.** You can configure a virtual server to use the L2 parameters to identify a connection.
- **ICMP Response**. You can configure the NetScaler to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.

  The following settings can be made on a virtual server:

  - When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.

- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

Configuring Sessionless Load Balancing Virtual Servers

When the NetScaler appliance performs load balancing, it creates and maintains sessions between clients and servers. The maintenance of session information places a significant load on the NetScaler resources, and sessions might not be needed in scenarios such as a direct server return (DSR) setup and the load balancing of intrusion detection systems (IDS). To avoid creating sessions when they are not necessary, you can configure a virtual server on the NetScaler for sessionless load balancing. In sessionless load balancing, the NetScaler carries out load balancing on a per-packet basis.

Sessionless load balancing can operate in MAC-based forwarding mode or IP-based forwarding mode.

For MAC-based forwarding, the IP address of the sessionless virtual server must be specified on all the physical servers to which the traffic is forwarded.

For IP-based forwarding in sessionless load balancing, the IP address and port of the virtual server need not be specified on the physical servers, because this information is included in the forwarded packets. When forwarding a packet from the client to the physical server, the NetScaler leaves client details such as IP address and port unchanged and adds the IP address and port of the destination.

Supported Setup

NetScaler sessionless load balancing supports the following service types and load balancing methods:

Service Types

- ANY for MAC-based redirection
- ANY, DNS, and UDP for IP-based redirection

Load Balancing Methods

- Round Robin
- Least Bandwidth
- LRTM (Least response time method)
- Source IP Hash
- Destination IP Hash

- Source IP Destination IP Hash
- Source IP Source Port Hash
- Custom Load

Limitations

Sessionless load balancing has the following limitations:

- The NetScaler must be deployed in two-arm mode.
- A service must be bound to only one virtual server.
- Sessionless load balancing is not supported for service groups.
- Sessionless load balancing is not supported for domain based services (DBS services).
- Sessionless load balancing in the IP mode is not supported for a virtual server that is configured as a backup to a primary virtual server.
- You cannot enable spillover mode.
- For all the services bound to a sessionless load balancing virtual server, the Use Source IP (USIP) option must be enabled.
- For a wildcard virtual server or service, the destination IP address will not be changed.

Note: While configuring a virtual server for sessionless load balancing, explicitly specify a supported load balancing method. The default method, Least Connection, cannot be used for sessionless load balancing.

Note: To configure sessionless load balancing in MAC-based redirection mode on a virtual server, the MAC-based forwarding option must be enabled on the NetScaler.

To add a sessionless virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a sessionless virtual server and verify the configuration:

- add lb vserver <vServerName> <serviceType> <ip> <port> **-m** <redirectionMode> **-sessionless** <(ENABLED|DISABLED)> **-lbMethod** <load_balancing_method>
- show lb vserver <vServerName>

Example

add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -lbMethod roundrobin -m ip

```
    Done
show lb vserver sesslessv1
    sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
    State: DOWN
    ...
    Effective State: DOWN
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    ...
    Persistence: NONE
    Sessionless LB: ENABLED
    Connection Failover: DISABLED
    L2Conn: OFF
    1) Policy : cmp_text Priority:8680 Inherited
    2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
```

To configure sessionless load balancing on an existing virtual server

At the NetScaler command prompt, type:

set lb vserver <vServerName> **-m** <redirectionMode> **-sessionless** <(ENABLED|DISABLED)> **-lbMethod** <load_balancing_method>

Example

```
set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
  Done
```

Parameters for configuring sessionless load balancing virtual servers

vServerName

The name of the virtual server that you are configuring.

m

The redirection mode that you want to use. MAC, IP.

sessionless

Perform load balancing on a per-packet basis, without establishing sessions. Possible values: ENABLED and DISABLED. Default: DISABLED.

lbMethod

The load balancing method. See Supported Setup.

To configure a sessionless virtual server by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, do one of the following:
   - To add a sessionless virtual server, click **Add**.
   - To specify sessionless load balancing for an existing virtual server, select it, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for Configuring Sessionless Load Balancing Virtual Servers" as shown:
   - Service Name*-serviceName
   - Protocol*-serviceType
   - Server*-serverName
   - Port*-port

   *A required parameter
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Methods and Persistence** tab, in the **LB Method** group, select a supported load balancing method.
5. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Advanced** tab, under **Redirection Mode**, select **MAC Based** or **IP Based**.
6. Click **Create**.
7. In the details pane, open the virtual server and verify the configuration.

Redirecting HTTP Requests to a Cache

The NetScaler cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the impact of responding to HTTP requests and improve your Web site performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the NetScaler appliance sends cacheable HTTP requests to the cache, and non-cacheable HTTP requests to the origin Web server. For more information on cache redirection, see Cache Redirection.

To configure cache redirection on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**cacheable** <Value>

Example

set lb vserver Vserver-LB-1 -cacheable yes

Parameters for configuring cache redirection

vServerName

The name of the virtual server that you are configuring.

cacheable

Virtual server requests to be routed to the cache redirection virtual server before sending them to the configured servers. Possible values: YES and NO. Default: NO.

To configure cache redirection on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, Vserver-LB-1), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. Select the **Cache Redirection** check box, and then click **OK**.

Directing Requests According to Priority

The NetScaler appliance supports prioritization of client requests with its priority queuing feature. This feature allows you to designate certain requests, such as those from important clients, as priority requests and sends them to the “front of the line,” so that the appliance responds to them first. This allows you to provide uninterrupted service to those clients through demand surges or DDoS attacks on your Web site.

For more information on priority queuing, see Priority Queuing.

To configure priority queuing on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**pq** <Value>

Example

set lb vserver Vserver-LB-1 -pq yes

Parameter for configuring priority queuing

vServerName

The name of the virtual server that you are configuring.

pq

Prioritizes client requests on the specified virtual server. Possible values: ON and OFF. Default: OFF.

To configure priority queuing on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, Vserver-LB-1), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. Select the **PQ** check box, and then click **OK**.

Note: You must configure priority queuing globally for it to function correctly.

Directing Requests to a Custom Web Page

The NetScaler appliance provides the SureConnect option to ensure that Web applications respond despite delays caused by limited server capacity or processing speed. SureConnect does this by displaying an alternative Web page of your choice when the server that hosts the primary Web page is either unavailable or responding slowly.

To configure SureConnect on a virtual server, you must first configure the alternative content. For information about configuring a SureConnect Web site, see the "Configuring SureConnect" chapter in the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681. After you configure the Web site, enable SureConnect on the load balancing virtual server to put your SureConnect custom Web page in use.

Note: For SureConnect to function correctly, you must configure it globally.

To enable SureConnect on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**sc** <Value>

Example

set lb vserver Vserver-LB-1 -sc yes

Parameters for configuring SureConnect

vServerName

The name of the virtual server that you are configuring.

sc

Assurance of a response from an application despite possible delays due to server capacity or processing speed. Possible values: ON and OFF. Default: OFF.

To enable SureConnect on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, Vserver-LB-1), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. Select the **SC** check box, and then click **OK**.

Enabling Delayed Cleanup of Virtual Server Connections

Under certain conditions, you can configure the **downStateFlush** setting to terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes and health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN.

A virtual server is marked DOWN only when all services bound to it are marked DOWN. When a virtual server goes DOWN, it terminates all connections, either immediately or after allowing existing connections to complete.

You must not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with two services bound to it, Service-TCP-1 and Service-TCP-2. The virtual server intercepts two connections, C1 and C2, and redirects them to Service-TCP-1 and Service-TCP-2, respectively. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-TCP-1	State of connections
E	E	Both client and server connections are terminated.
E	D	Both client and server connections are terminated. In case of HTTP services, both client and server connections are terminated only if the transaction is active. If the transaction is not active, only client connections are terminated.

D	E	Both client and server connections are terminated. In case of HTTP services, both client and server connections are terminated only if the transaction is active. If the transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

Note: In case of HTTP services, the downStateFlush setting is effective only when the client is connected to the server.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown of a service, see Graceful Shutdown of Services.

To configure the down state flush setting on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**downStateFlush** <Value>

Example

set lb vserver Vserver-LB-1 -downStateFlush enabled

Parameters for configuring down state flush

vServerName

The name of the virtual server that you are configuring.

downStateFlush

State that performs delayed cleanup of connections on the virtual server. Possible values: ENABLED and DISABLED. Default: ENABLED

To configure the down state flush setting on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. Select the **Down state flush** check box, and then click **OK**.

Graceful Shutdown of Services

During scheduled network outages such as system upgrades or hardware maintenance, you may have to close or disable some services. To avoid disrupting sessions that have already been established, you can specify a wait time, which places a service in the transition out of service (TROFS) state until the specified wait time expires. The service then enters the out of service (OFS) state.

Often, however, you cannot estimate the amount of time needed for all the connections to a service to complete the existing transactions. If a transaction is unfinished when the wait time expires, shutting down the service may result in data loss. In this case, you can specify graceful shutdown for the service, so that the service is disabled only when all the established connections are closed by either the server or the client.

Persistence is maintained according to the specified method even if you enable graceful shutdown. The system continues to serve all the persistent clients, including new connections from the clients, unless the service is marked DOWN during the graceful shutdown state as a result of the checks made by a monitor.

The following table describes graceful shutdown options.

Table 1. Graceful Shutdown Options

State	Results
Graceful shutdown is enabled and a wait time is specified.	Service is shut down after the last of the previously established connections is served, even if the wait time has not expired. The appliance checks the status of the connections once every second. If the wait time expires, any open sessions are closed.
Graceful shutdown is disabled and a wait time is specified.	Service is shut down only after the wait time expires, even if all established connections are served before expiration.
Graceful shutdown is enabled and no wait time is specified.	Service is shut down only after the last of the previously established connections is served, regardless of the time taken to serve the last connection.
Graceful shutdown is disabled and no wait time is specified.	No graceful shutdown. Service is shut down immediately after the disable option is chosen or the disable command is issued. (The default wait time is zero seconds.)

To terminate existing connections when a service or a virtual server is marked DOWN, you can use the Down State Flush option. For more information, see Enabling Delayed Cleanup of Virtual Server Connections.

To configure graceful shutdown for a service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to shut down a service gracefully and verify the configuration:

- disable service <serviceName> [<delayInSeconds>] [**-graceFul** (YES|NO)]
- show service <serviceName>

Example

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
1) Monitor Name: tcp-default
State: UP           Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN          Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
```

Parameters for configuring a graceful shutdown for a service

serviceName

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

delayInSeconds

The time in seconds after which the service is marked as OUT OF SERVICE.

graceFul

Wait for all previously established connections to this service to be closed before disabling the service. Possible values: YES, NO. Default: NO.

To configure graceful shutdown for a service by using the NetScaler configuration utility

1. In the **navigation** pane, expand **Load Balancing** and then click **Services**.
2. In the **details** pane, select the service, and then click **Disable**.
3. To delay disabling the service, in the **Wait Time** dialog box, type the wait time after which the service is to be disabled.
4. To disable the service only after all previously initiated transactions have been completed, check the **Graceful Shutdown** check box.
5. Click **Enter.**
6. In the **Services** pane, you can verify that the service is marked as UP until the wait time expires and after that, it is marked as OUT OF SERVICE.

Rewriting Ports and Protocols for HTTP Redirection

Virtual servers and the services that are bound to them may use different ports. When a service responds to an HTTP connection with a redirect, you may need to configure the NetScaler appliance to modify the port and the protocol to ensure that the redirection goes through successfully. You do this by enabling and configuring the **redirectPortRewrite** setting.

This setting affects HTTP traffic only. When this setting is enabled on a virtual server, the virtual server rewrites the port on redirects, replacing the port used by the service with the port used by the virtual server.

This setting can be used in the following scenarios:

- The virtual server is of type HTTP and the services are of type SSL.
- The virtual server is of type SSL and the services are of type HTTP.
- The virtual server port is different than the service port.

When requests are of type HTTP and services are of type SSL, the virtual server rewrites the port of the HTTP requests (usually port 80) to that of SSL (usually port 443) before it forwards those requests. Then, the virtual server rewrites the port of the HTTPS responses from those services to that of HTTP before it forwards the responses to the client. The following table summarizes this process.

Redirect URL	URL after port rewrite
Case 1 - Redirect port rewrite enabled and virtual server on port 80	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/
Case 2- Redirect port rewrite enabled and virtual server on port 8080.	
http://domain.com/	http://domain.com:8080/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

When requests are of type SSL and services are of type HTTP, the virtual server rewrites the port of the SSL requests (usually port 443) to that of HTTP (usually port 80) before it forwards those requests. Then, the virtual server rewrites the port of the HTTP responses to

that of HTTPS before it forwards them to the client.

When both requests and responses are of same type, the virtual server leaves the port unchanged. For more information about SSL redirects, see SSL Offload and Acceleration.

To configure HTTP redirection on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**redirectPortRewrite** <Value>

Example

set lb vserver Vserver-LB-1 -redirectPortRewrite enabled

Parameters for redirect port rewrite

vServerName

The name of the virtual server that you are configuring.

redirectPortRewrite

State of port rewrite while performing HTTP redirect. Possible values: ENABLED and DISABLED. Default: DISABLED.

To configure HTTP redirection on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. Select the **Redirect Port Rewrite** check box, and then click **OK**.

Inserting the IP Address and Port of a Virtual Server in the Request Header

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the NetScaler appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, then you must configure the required header tag on both virtual servers.

Note: This option is not supported for wildcarded virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**insertVserverIPPort** <vServerIPAddress>

Example

set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr

Parameters for virtual server IP port insertion

vServerName

The name of the virtual server that you are configuring.

insertVserverIPPort

Virtual IP address and port header insertion option for the virtual server.

VIPADDR-Header contains the virtual server IP address and port number without any translation.

If VIPADDR is not specified, the header is inserted with the name specified in the default header tag vip-header and the virtual server IP and port are inserted in the request with the default header tag vipHeader.

If VIPADDR is specified, the header is inserted with the user-specified name in vipHeader. The virtual server IP and port are inserted in the request with the user-specified header tag vipHeader.

OFF- The virtual IP and port header insertion option is disabled. The virtual server and port number are not inserted.

V6TOV4MAPPING - If the virtual server uses an IPv6 address and the server uses IPv4, this setting maps the virtual server address and port to the IPv4 address.

Possible values: OFF, VIPADDR, and V6TOV4MAPPING. Default: OFF.

To insert the IP address and port of the virtual server in the client requests by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. In the **Vserver IP Port Insertion** list, select the **VIPADDR** or **V6TOV4MAPPING**, and then type the port header in a text box next to **Vserver IP Port Insertion** box.
5. Click **OK**.

Using a Specified Source IP for Backend Communication

For communication with the physical servers or other peer devices, the NetScaler appliance uses an IP address owned by it as the source IP address. NetScaler maintains a pool of subnet IP addresses (SNIPs) and mapped IP addresses (MIPs), and dynamically selects an IP address while connecting with a server. Depending on the subnet in which the physical server is placed, NetScaler decides whether a MIP should be used or SNIP. This address pool is used for sending traffic as well as monitor probes.

In many situations, you may want the NetScaler to use a specific IP address or any IP address from a specific set of IP addresses for backend communications. The following are a few examples:

- A server can distinguish monitor probes from traffic if the source IP address used for monitor probes belongs to a specific set.
- To improve server security, a server may be configured to respond to requests from a specific set of IP addresses or, sometimes, from a single specific IP address. In such a case, the NetScaler can use only the IP addresses accepted by the server as the source IP address.
- The NetScaler can manage its internal connections efficiently if it can distribute the MIPs or SNIPs into IP sets and use an address from a set only for connecting to a specific service.

To configure the NetScaler to use a specified source IP address, create net profiles (network profiles) and configure the NetScaler entities to use the profile. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. A net profile has a SNIP or MIP address that can be used as the source IP address. It can be a single IP address or a set of IP addresses, referred to as an IP set. If a net profile has an IP set, NetScaler dynamically selects an IP address from the IP set at the time of connection. If a profile has a single IP address, the same IP address is used as the source IP.

If a net profile is bound to a load balancing or content switching virtual server, the profile will be used for sending traffic to all the services bound to it. If a net profile is bound to a service group, NetScaler uses the profile for all the members of the service group. If a net profile is bound to a monitor, NetScaler uses the profile for all the probes sent from the monitor.

Note: The user-specified source IP feature is supported only on NetScaler 9.3.e.

Usage of a net profile for sending traffic:

If the **Use Source IP Address (USIP)** option is enabled, NetScaler uses the IP address of the client and ignores all the net profiles. If the **USIP** option is not enabled, NetScaler selects the source IP in the following manner:

- If there is no net profile on the virtual server or the service/service group, NetScaler uses the default method.
- If there is a net profile only on the service/service group, NetScaler uses that net profile.
- If there is a net profile only on the virtual server, NetScaler uses the net profile.
- If there is a net profile both on the virtual server and service/service group, NetScaler uses the net profile bound to the service/service group.

Usage of a net profile for sending monitor probes:

For monitor probes, NetScaler selects the source IP in the following manner:

- If there is a net profile bound to the monitor, NetScaler uses the net profile of the monitor. It ignores the net profiles bound to the virtual server or service/service group.
- If there is no net profile bound to the monitor,
  - If there is a net profile on the service/service group, NetScaler uses the net profile of the service/service group.
  - If there is no net profile even on the service/service group, NetScaler uses the default method of selecting a source IP.

Note: If there is no net profile bound to a service, NetScaler looks for a net profile on the service group if the service is bound to a service group.

To use a specified source IP address for communication, go through the following steps:

1. Create IP sets from the pool of SNIPs and MIPs owned by the NetScaler. An IP set can consist of both MIP and SNIP addresses. For instructions, see Creating IP Sets.
2. Create net profiles. For instructions, see Creating a Net Profile.
3. Bind the net profiles to NetScaler entities. For instructions, see Binding a Net Profile to a NetScaler Entity.

Note: A net profile can have only the IP addresses specified as SNIP or MIP on the NetScaler.

Managing Net Profiles

A net profile (or network profile) contains an IP address or an IP set. During communication with physical servers or peers, the NetScaler appliance uses the addresses specified in the profile as the source IP address. For more information on the use of net profiles, see Using a User-specified Source IP Address for Backend Communication.

Note: The net profile feature is supported only on NetScaler 9.3.e.

- For instructions on creating a network profile, see Creating a Network Profile.
- For instructions on binding a network profile to a NetScaler entity, see Binding a Network Profile.

Creating an IP Set

An IP set is a set of IP addresses, which are configured on the NetScaler appliance as Subnet IP addresses (SNIPs) or Mapped IP addresses (MIPs). An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it. To create an IP set, add an IP set and bind NetScaler owned IP addresses to it. Both MIP addresses and SNIP addresses can be present in the same IP set. For more information about the use of IP sets, see Using a User-specified Source IP Address for Backend Communication.

Note: The IP Set feature is supported only on NetScaler 9.3.e.

Note: Before binding an IP address to a set, make sure that the IP address has been added as a NetScaler MIP or SNIP address. For more information, see Configuring Mapped IP Addresses (MIPs) or Configuring Subnet IP Addresses (SNIPs). Configuring Mapped IP Addresses (MIPs) or Configuring Subnet IP Addresses (SNIPs).

To create an IP set by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add ipset <name>
- bind ipset <name> <ipAddress>
  or
- bind ipset <name> < ipAddressRange>
- show ipset [<name>]
  The above command shows the names of all the IP sets on the NetScaler if you do not pass any name. It shows the IP addresses bound to the specified IP set if you pass a name.

Examples

1.

```
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done
```

2.

```
 > add ipset testnwipset
 Done
> bind ipset testnwipset 21.21.21.[21-25]
```

IPAddress "21.21.21.21" bound
IPAddress "21.21.21.22" bound
IPAddress "21.21.21.23" bound
IPAddress "21.21.21.24" bound
IPAddress "21.21.21.25" bound
Done

3.
> bind ipset skpipset 11.11.11.101
ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address owned by the NetScaler]
> add ns ip 11.11.11.101 255.255.255.0 -type SNIP
ip "11.11.11.101" added
Done
> bind ipset skpipset 11.11.11.101
IPAddress "11.11.11.101" bound
Done

4.
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
Done

5.
> sh ipset skpnewipset
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
Done

Parameters for configuring an IP set

name (Name)

The name of the IP set. The name can have up to 127 characters. It must begin with an alphanumeric character or underscore, and must contain only alphanumerics, '_', '#', '.', ' ', ':', '@', '=' or '-'.

ipAddress (IP Address)

A SNIP or MIP address on the NetScaler.

ipAddressRange

A contiguous range of SNIP or MIP address on the NetScaler.

To create an IP set by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **IP Sets**.
2. In the details pane, do one of the following:
   - To create a new IP set, click **Add**.
   - To modify an existing IP set, select the IP set, and then click **Open**.
3. In the **Create IP Set** dialog box, set the following parameters:
   - Name
   - IP Address (The MIPs and SNIPs specified on the NetScaler are displayed. Check the IP addresses that you want to bind to the IP set. You can select more than one.)

   If you want to add an IP address to the pool of MIPs or SNIPs of the NetScaler, do one of the following:
   - To add an IPv4 address, click **Add IPv4**, and then in the **Create IP** dialog box, type the necessary details.
   - To add an IPv6 address, click **Add IPv6**, and then in the **Create IPV6** dialog box, type the necessary details.
4. Click **Create**.

Creating a Net Profile

A net profile (network profile) consists of one or more MIP or SNIP addresses of the NetScaler. For more information about the usage of net profiles, see Using a User-specified Source IP Address for Backend Communication.

To create a net profile by using the NetScaler command line

At the NetScaler command prompt, type:

add netprofile <name> [**-srcIp** <srcIpVal>] If the srcIpVal is not provided in this command, it can be provided later by using the set netprofile command.

Examples

```
> add netprofile skpnetprofile1 -srcIp 21.21.20.1
Done

> add netprofile baksnp -srcIp bakipset
Done

> set netprofile yahnp -srcIp 12.12.23.1
```

```
Done

> set netprofile citkbnp -srcIp citkbipset
Done
```

Parameters for creating a net profile

name (Name)

The name of the net profile. The name can have up to 127 characters. It must begin with an alphanumeric character or underscore, and must contain only alphanumerics, '_', '#', '.', ' ', ':', '@', '=' or '-'.

srcIpVal (IP address or IP Set)

IP address or the name of an IP set.

To create a net profile by using the NetScaler configuration utility

1. In the navigation pane, expand **Network**, and then click **Net Profiles**.
2. In the **Create Net Profile** dialog box, type a name for the net profile.
3. To specify the source IP address, do one of the following:
   - Check **IP Address** and select an IP address from the **IP Address** drop-down list.
   - Check **IP Set** and select the name of an IP set from the **IP Set** drop-down list.

   If you want to add a new IP address or IP set, click **New** and type the necessary details in the dialog box that is displayed. If you want to modify an entry after selecting it from the drop-down list, click **Modify** and change the values. To unbind an IP address or IP set from a net profile, select the blank entry from the drop down list, and then click **OK**.
4. Click **Create**.

Binding a Net Profile to a NetScaler Entity

A net profile can be bound to a load balancing virtual server, service, service group, or a monitor. For more information about the effect of binding a net profile to a NetScaler entity, see Using a User-specified Source IP Address for Backend Communication.

Note: You can bind a net profile at the time of creating a NetScaler entity or bind it to an already existing entity.

To bind a net profile to a server by using the NetScaler command line

You can bind a net profile to load balancing virtual servers and content switching virtual servers. Specify the appropriate virtual server.

At the NetScaler command prompt, type:

- set lb vserver <vserver_name> **-netProfile** <net_profile_name>
  or
- set cs vserver <vserver_name> **-netProfile** <net_profile_name>

Examples

```
set lb vserver skpnwvs1 -netProfile gntnp
 Done
set cs vserver mmdcsv -netProfile mmdnp
 Done
```

To bind a net profile to a virtual server by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing** or **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind a net profile, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, or **Configure Virtual Server (Content Switching)** click the **Profiles** tab.
4. In the **Net Profile** drop-down list, select a net profile. In this dialog box, you can click **New…** to add a net profile or **Modify…** to modify the selected net profile.
5. Click **OK**.

To bind a net profile to a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <service_name> **-netProfile** <net_profile_name>

Example

```
set service brnssvc1 -netProfile brnsnp
 Done
```

To bind a net profile to a service by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service to which you want to bind a net profile, and then click **Open**.
3. In the **Configure Service** dialog box, click the **Profiles** tab.
4. In the **Net Profile** drop-down list, select a net profile. In this dialog box, you can click **New…** to add a net profile or **Modify…** to modify the selected net profile.
5. Click **OK**.

To bind a net profile to a service group by using the NetScaler command line

At the NetScaler command prompt, type:

set servicegroup <servicegroup_name> **-netProfile** <net_profile_name>

Example

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
```

To bind a net profile to a service group by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group to which you want to bind a net profile, and then click **Open**.
3. In the **Configure Service Group** dialog box, click the **Profiles** tab.
4. In the **Net Profile** drop-down list, select a net profile. In this dialog box, you can click **New…** to add a net profile or **Modify…** to modify the selected net profile.
5. Click **OK**.

To bind a net profile to a monitor by using the NetScaler command line

At the NetScaler command prompt, type:

set monitor <monitor_name> **-netProfile** <net_profile_name>

Example

```
set monitor brnsecvmon1 -netProfile brnsmonnp
 Done
```

To bind a net profile to a monitor by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, select the monitor to which you want to bind a net profile, and then click **Open**.
3. In the **Configure Monitor** dialog box, in the **Net Profile** drop-down list, select a net profile.
4. Click **OK**.

Setting a Timeout Value for Idle Client Connections

You can configure a virtual server to terminate any idle client connections after a configured timeout period elapses. When you configure this setting, the NetScaler appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection.

To set a time-out value for idle client connections by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**cltTimeout** <Value>

Example

set lb vserver Vserver-LB-1 -cltTimeout 100

Parameters for setting the client time-out value

vServerName

The name of the virtual server that you are configuring.

cltTimeout

Idle time (in seconds) after which the client connection is terminated. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

To set a time-out value for idle client connections by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. In the **Client Time-out (secs)** text box, type the timeout value (for example, **100**).
5. Click **OK**.

Managing RTSP Connections

The NetScaler appliance can use either of two topologies—*NAT-on mode* or *NAT-off mode*—to load balance RTSP servers. In NAT-on mode, Network Address Translation (NAT) is enabled and configured on the appliance. RTSP requests and responses both pass through the appliance. You must therefore configure the appliance to perform network address translation (NAT) to identify the data connection.

For more information about enabling and configuring NAT, see the "IP Addressing" chapter in the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

In NAT-off mode, NAT is not enabled and configured. The appliance receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. The load balanced RTSP servers send their responses directly to the client, bypassing the appliance. You must therefore configure the appliance to use Direct Server Return (DSR) mode, and assign publicly accessible FQDNs in DNS to your load balanced RTSP servers.

For more information about enabling and configuring DSR mode, see the Configuring Load Balancing in Direct Server Return Mode. For more information about configuring DNS, see Domain Name System.

In either case, when you configure RTSP load balancing, you must also configure rtspNat to match the topology of your load balancing setup.

To configure RTSP NAT by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**RTSPNAT** <ValueOfRTSPNAT>

Example

set lb vserver vserver-LB-1 -RTSPNAT ON

Parameters for configuring RTSP

vServerName

The name of the virtual server that you are configuring.

rtspNat

Whether the appliance is configured to use NAT or not when load balancing RTSP services. When the appliance is configured for the NAT-on mode, you must set rtspNat ON. When the NetScaler is configured for NAT-off mode, you must set rtspNat OFF. Possible values: ON and OFF. Default value: OFF.

To configure RTSP NAT by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure sessionless load balancing (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. Select the **RTSP Natting** check box, and then click **OK**.

Managing Client Traffic on the Basis of Traffic Rate

You can monitor the rate of traffic that flows through load balancing virtual servers and control the behavior of the NetScaler appliance based on the traffic rate. You can throttle the traffic flow if it is too high, cache information based on the traffic rate, and if the traffic rate is too high redirect excess traffic to a different load balancing virtual server. You can apply rate-based monitoring to HTTP and Domain Name System (DNS) requests.

For more information on rate-based policies, see the *Citrix NetScaler AppExpert Guide* at http://support.citrix.com/article/CTX128682.

Identifying a connection with Layer 2 Parameters

Generally, to identify a connection, the NetScaler uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.

Enabling the L2Conn parameter for a load balancing virtual server allows multiple TCP and non-TCP connections with the same 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) to co-exist on the NetScaler appliance. The appliance uses both the 4-tuple and the Layer 2 parameters to identify TCP and non-TCP connections.

You can enable the L2Conn option in the following scenarios:

- Multiple VLANs are configured on the NetScaler appliance, and a firewall is set up for each VLAN.
- You want the traffic originating from the servers in one VLAN and bound for a virtual server in another VLAN to pass through the firewalls configured for both VLANs.

Note: In NetScaler 9.3 Classic, the **L2 Connection** option is not supported for load balancing virtual servers. It is supported only for cache redirection virtual servers. In NetScaler 9.3 nCore, the **L2 Connection** option is supported for both load balancing and cache redirection virtual servers.

Therefore, when an nCore NetScaler appliance on which the l2Conn parameter is set for one or more load balancing virtual servers is downgraded to a Classic build or to an nCore build that does not support the l2Conn parameter, the load balancing configurations that use the l2Conn parameter become ineffective.

To configure the L2 connection option by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <name> <serviceType> <ip> <port> **-l2Conn** ON

Example

add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON

Parameters for configuring a virtual server

vServerName

Name of the virtual server that is associated with the service. The name must not exceed 127 characters, and the leading character must be a number or a letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ipAddress

IP address of the virtual server, in either IPv4 or Ipv6 format.

serviceType

The type of services to which the virtual server distributes requests.

port

Port on which the virtual server listens for client connections.

l2Conn

The tuple used to identify a connection includes the layer 2 parameters.

To configure the L2 connection option by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers** .
2. In the details pane, click **Add**.
3. In the **CreateVirtual Server (Load Balancing)** dialog box, specify values which correspond to parameters described in "Parameters for configuring a virtual server" as shown
   - Name*-vServerName
   - IP Address*-ipAddress
   - Protocol*-serviceType
   - Port*-port

   *A required parameter
4. On the **Advanced** tab, select the **L2 Connection** check box.
5. Click **Create.**
6. Open the virtual server you configured and verify the configuration.

Configuring the Prefer Direct Route Option

On a wildcard load balancing virtual server if you explicitly configure a route to a destination, by default, the NetScaler appliance forwards traffic according to the configured route. If you want the NetScaler to not look up for the configured route, you can set the Prefer Direct Route option to NO.

If a device is directly connected to a NetScaler appliance, the NetScaler directly forwards traffic to the device. For example, if the destination of a packet is a firewall, the packet need not be routed through another firewall. However, in some cases, you may want the traffic to go through the firewall even if the device is directly connected to it. In such cases, you can set the Prefer Direct Route Option to NO.

Note: The preferDirectRoute setting is applicable to all the wildcard virtual servers on the NetScaler appliance.

To set the prefer direct route option by using the NetScaler command line

At the NetScaler command prompt, type:

set lb parameter -preferDirectRoute (YES | NO)

Example

set lb parameter -preferDirectRoute YES

Parameter for configuring prefer direct route option

preferDirectRoute

If enabled, the NetScaler looks up for the configured route. Possible values: YES, NO. Default: YES.

To set the prefer direct route option by using the NetScaler configuration utility

1. In the navigation pane, click **Load Balancing**.
2. Under **Settings**, click **Configure Load Balancing Parameters**.
3. In the **Configure Load Balancing Parameters** dialog box, select the **Prefer Direct Route** check box.
4. Click **OK**.

Advanced Load Balancing Settings

In addition to configuring virtual servers, you can configure advanced settings for services.

The No-Monitor Option for Services

If you use an external system to perform health checks on the services and do not want the NetScaler appliance to monitor the health of a service, you can set the no-monitor option for the service. If you do so, the appliance does not send probes to check the health of the service but shows the service as UP. Even if the service goes DOWN, the appliance continues to send traffic from the client to the service as specified by the load balancing method.

The monitor can be in the ENABLED or DISABLED state when you set the no-monitor option, and when you remove the no-monitor option, the earlier state of the monitor is resumed.

You can set the no-monitor option for a service when creating the service. You can also set the no-monitor option on an existing service.

The following are the consequences of setting the no-monitor option:

- If a service for which you enabled the no-monitor option goes down, the NetScaler continues to show the service as UP and continues to forward traffic to the service. A persistent connection to the service can worsen the situation. In that case, or if many services shown as UP are actually DOWN, the system may fail. To avoid such a situation, when the external mechanism that monitors the services reports that a service that is DOWN, remove the service from the NetScaler configuration.
- If you configure the no-monitor option on a service, you cannot configure load balancing in the Direct Server Return (DSR) mode. For an existing service, if you set the no-monitor option, you cannot configure the DSR mode for the service.

To set the no-monitor option for a new service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a service with the health monitor option, and verify the configuration:

add service <serviceName> <ipAddress | serverName> <serviceType> <port> **-healthMonitor** (YES|NO)

Example

```
>add service nomonsrvc 10.102.21.21 http 80
-healthMonitor no
Done
> show service nomonsrvc
```

```
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done
```

To set the no-monitor option for an existing service by using the NetScaler command line

At the NetScaler command prompt, type the following command to set the health monitor option:

set service <serviceName> **-healthMonitor** (YES|NO)

Example

By default, the state of a service and the state of the corresponding monitor are UP.

```
 >show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP


1) Monitor Name: http-ecv
 State: UP Weight: 1
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done
```

When the no-monitor option is set on a service, the state of the monitor changes to UNKNOWN.

```
> set service LB-SVC1 -healthMonitor NO
Done
> show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
```

```
State: UP
Last state change was at Fri Dec 10 10:17:37 2010.
Time since last state change: 5 days, 18:55:48.710
Health monitoring: OFF

1) Monitor Name: http-ecv
   State: UNKNOWN Weight: 1
        Probes: 100028 Failed [Total: 0 Current: 0]
   Last response: Probe skipped - Health monitoring is turned off.
     Response Time: 0.0 millisec
  Done
When the no-monitor option is removed, the earlier state of the monitor is resumed.
> set service LB-SVC1 -healthMonitor YES
Done
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010
Time since last state change: 5 days, 18:57:47.880
1) Monitor Name: http-ecv
   State: UP Weight: 1
   Probes: 100029 Failed [Total: 0 Current: 0]
   Last response: Success - Pattern found in response.
   Response Time: 5.690 millisec
   Done
```

Parameters for configuring a service

serviceName

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ipAddress

IP address of the server that is associated with the service, in either IPv4 or IPv6 format.

serverName

Name of the server that is associated with the service. The name must not exceed 127 characters, and the leading character must be a number or a letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serviceType

Protocol supported by the service. Possible values: HTTP, FTP, TCP, UDP, SSl, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, DNS_TCP, ADNS_TCP, RDP, RADIUS.

port

The port number used for the service.

healthMonitor

The monitoring option for the service. Possible values: YES, NO. Default: YES.

To set the no-monitor option for a service by using the NetScaler configuration utility

1. In the navigation pane, click **Load Balancing** and then click **Services.**
2. In the details pane, do one of the following:
   - To create a new service, click **Add**.
   - To modify an existing service, select the service and then click **Open**.
3. In the **Create Service** or **Configure Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring a service” as shown:
   - **Service Name***-serviceName
   - **Protocol***-serviceType
   - **Server***-ipAddress
   - **Port***-port
   - **Health Monitor**—healthMonitor

   *A required parameter
4. Click **Create**. The service you created appears in the **Services** pane.
5. From the **services** pane, open the service you added, and verify the health monitor setting.

Protecting Applications on Protected Servers Against Traffic Surges

The NetScaler provides the surge protection option to maintain the capacity of a server or cache. The NetScaler regulates the flow of client requests to servers and controls the number of clients that can simultaneously access the servers. The NetScaler blocks any surges passed to the server, thereby preventing overloading of the server.

For more information about surge protection, see Surge Protection.

To set surge protection on the service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**sp** <Value>

Example

set service Service-HTTP-1 -sp ON

Parameters for configuring surge protection on the service

ServiceName

The name of the service that you are configuring.

sp

State of surge protection for the service. Possible values: ON and OFF. Default: OFF.

To set surge protection on the service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure surge protection (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab, scroll down, and under **Others**, select the **Surge Protection** check box.
4. Click **OK**.

Note: For surge protection to function correctly, you must enable it globally. For more information about configuring surge protection globally, see Surge Protection.

Enabling Delayed Cleanup of Service Connections

When delayed cleanup of service connections is enabled, the NetScaler performs a delayed cleanup of the connections on a service that is down. This setting is described in Enabling Delayed Cleanup of Virtual Server Connections.

To set down state flush on the service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**downStateFlush** <Value>

Example

set service Service-HTTP-1 -downStateFlush enabled

Parameters for configuring down state flush on the service

ServiceName

The name of the service that you are configuring.

downStateFlush

Delayed clean up of connections on this service. Possible values: ENABLED and DISABLED. Default: ENABLED.

To set down state flush on the service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure down state flush (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Scroll down, and under **Others**, select the **Down state flush** check box.
5. Click **OK**.

Directing Requests to a Custom Web Page

The NetScaler provides the SureConnect option to ensure the response from an application. For more information about the

SureConnect option, see Sure Connect.

To set SureConnect on the service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**sc** <Value>

Example

set service Service-HTTP-1 -sc ON

Parameters for configuring SureConnect on the service

ServiceName

The name of the service that you are configuring.

sc

State of SureConnect for the service. This parameter is supported for legacy purposes only. It has no effect on the NetScaler, and its only valid value is OFF. Possible values: ON and OFF. Default: OFF.

To set SureConnect on the service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure SureConnect (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Scroll down, and under **Others**, select the **Sure Connect** check box.
5. Click **OK**.

Note: For SureConnect to function correctly, you must set it globally. For more information about configuring SureConnect globally, see the "Configuring SureConnect" chapter in the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Enabling Access to Services When Down

You can enable access to a service when it is disabled or in a DOWN state by configuring the NetScaler appliance to use Layer 2 mode to bridge the packets sent to the service. Normally, when requests are forwarded to services that are DOWN, the request packets are dropped. When you enable the Access Down setting, however, these request packets are sent directly to the load balanced servers.

For more information about Layer 2 and Layer 3 modes, see the "IP Addressing" chapter in the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

For the appliance to bridge packets sent to the DOWN services, enable Layer 2 mode with the **accessDown** parameter.

To enable access down on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> **-accessDown** <Value>

Example

set service Service-HTTP-1 -accessDown YES

Parameters for configuring Access Down

ServiceName

The name of the service that you are configuring.

accessDown

Access to disabled or DOWN services. If this option is enabled, and the service goes DOWN, all packets to the service are bridged. If this option is disabled, and the service goes DOWN, the packets are dropped. Possible values: YES and NO. Default: NO.

To enable access down on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure access down (for example, **Service-HTTP-1**), click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Scroll down, and under **Others**, select the **Access Down** check box.
5. Click **OK**.

Enabling TCP Buffering of Responses

The NetScaler appliance provides a TCP buffering option that buffers only responses from the load balanced server. This enables the appliance to deliver server responses to the client at the maximum speed that the client can accept them. The appliance allocates from 0 through 4095 megabytes (MB) of memory for TCP buffering, and from 4 through 20480 kilobytes (KB) of memory per connection.

To enable TCP Buffering on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**TCPB** <Value>

Example

set service Service-HTTP-1 -TCPB YES

Parameters for configuring TCP buffering

ServiceName

The name of the service that you are configuring.

TCPB

State of the TCP buffering feature for the service.Possible values: YES and NO. Default: NO.

To enable TCP Buffering on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure TCP buffering (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Scroll down, and under **Settings**, select the **TCP Buffering** check box.
5. Click **OK**.

Note: TCP buffering set at the service level takes precedence over the global setting. For more information about configuring TCP buffering globally, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Enabling Compression

The NetScaler appliance provides a compression option to transparently compress HTML and text files by using a set of built-in compression policies. Compression reduces bandwidth requirements and can significantly improve server responsiveness in bandwidth-constrained setups. The compression policies are associated with services bound to the virtual server. The policies determine whether a response can be compressed and send compressible content to the appliance, which compresses it and sends it to the client.

To enable compression on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**CMP** <Value>

Example

set service Service-HTTP-1 -CMP YES

Parameters for configuring compression

ServiceName

The name of the service that you are configuring.

CMP

State of the HTTP compression feature for the service. Possible values: YES, NO. Default: NO.

To enable compression on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure compression (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Settings**, select the **Compression** check box.
5. Click **OK**.

Note: For compression to function correctly, you must enable it globally. For more information about configuring compression globally, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Maintaining Client Connection for Multiple Client Requests

You can set the client keep-alive parameter to configure an HTTP or SSL service to keep a client connection to a Web site open across multiple client requests. If client keep-alive is enabled, even when the load balanced Web server closes a connection, the NetScaler appliance keeps the connection between the client and itself open. This setting allows services to serve multiple client requests on a single client connection.

If you do not enable this setting, the client will open a new connection for every request that it sends to the Web site. The client keep-alive setting saves the packet round trip time required to establish and close connections. This setting also reduces the time to complete each transaction. Client keep-alive can be enabled only on HTTP or SSL service types.

For more information about client keep-alive, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

To enable client keep-alive on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**CKA** <Value>

Example

set service Service-HTTP-1 -CKA YES

Parameters to configure client keep-alive

ServiceName

The name of the service that you are configuring.

CKA

State of the Client Keep-Alive feature. Possible values: YES, NO. Default: NO.

To enable client keep-alive on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure client keep-alive (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Settings**, select the **Client Keep-Alive** check box.
5. Click **OK**.

Note: Client keep-alive set at the service level takes precedence over the global client keep-alive setting. For more information about configuring Client keep-alive globally, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681.

Inserting the IP Address of the Client in the Request Header

A NetScaler uses the mapped IP address (MIP) to connect to the server. The server need not be aware of the client.

However, in some situations, the server needs to be aware of the client it has to serve. When you enable the client IP setting, the NetScaler inserts the client's IPv4 or IPv6 address while forwarding the requests to the server. The server inserts this client IP in the header of the responses. The server is thus aware of the client.

To insert client IP address in the client request by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**CIP** <Value> <cipHeader>

Example

set service Service-HTTP-1 -CIP enabled X-forwarded-for

Parameters for inserting client IP address in the client request

ServiceName

The name of the service that you are configuring.

CIP

Client IP address header addition option for the service. Possible values: ENABLED and DISABLED. Default: DISABLED.

This option works with IPv4 and IPv6 addresses.

cipHeader

The name of the HTTP header that the NetScaler inserts and to which it adds the IP address of the client as the header value. If client IP insertion is enabled, and the client IP header is not specified, then the NetScaler sets the client IP header. The default is blank (NetScaler uses a blank HTTP header).

To insert client IP address in the client request by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to add the client IP address in the request (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Settings**, select the **Client IP** check box.
5. In the **Header** text box, type the header tag (for example, **X-Forwarded-for**).
6. Click **OK**.

Using the Source IP Address of the Client When Connecting to the Server

You can configure the NetScaler appliance to forward packets from the client to the server without changing the source IP address. This is useful when you cannot insert the client IP address into a header, such as when working with non-HTTP services.

For more information about configuring USIP globally, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

For information about using the port of the client when connecting to the server, see Using the Client Port When Connecting to the Server.

To enable USIP mode for a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**usip** (YES | NO)

Example

set service Service-HTTP-1 -usip YES

Parameters to configure USIP mode

ServiceName

The name of the service that you are configuring.

usip

Determines the source IP address used when the NetScaler appliance connects to the server. If this option is set to YES, the NetScaler uses the client IP address. Possible values: YES, NO. Default: NO.

Note: USIP does not work when you bind an IPv6 service with USIP enabled to an IPv4 virtual server, or when you bind an IPv4 service with USIP enabled to an IPv6 virtual server.

To enable USIP mode for a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to enable the USIP mode, and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Settings**, select the **Use Source IP** check box.
5. Click **OK**.

Using the Client Port When Connecting to the Server

When the NetScaler appliance connects to a physical server, it can use the source port from client's request, or it can use a proxy port as the source port for the connection. You can set the **Use Proxy Port** parameter to YES to handle situations such as the following scenario:

- The NetScaler appliance is configured with two load balancing virtual servers, LBVS1 and LBVS2.
- Both the virtual servers are bound to the same service, S-ANY.
- Use (the client's) source IP address (USIP) is enabled on the service.
- Client C1 sends two requests, Req1 and Req2, for the same service.
- Req1 is received by LBVS1 and Req2 is received by LBVS2.
- LBVS1 and LBVS2 forward the request to S-ANY, and when S-ANY sends the response, they forward the response to the client.
- Consider two cases:
    - Use the client port. When the NetScaler uses the client port, both the virtual servers use the client's IP address (because USIP is ON) and the client's port when connecting to the server. Therefore, when the service sends the response, the NetScaler cannot determine which virtual server should receive the response.
    - Use proxy port. When the NetScaler uses a proxy port, the virtual servers use the client's IP address (because USIP is ON), but different ports when connecting to the server. Therefore, when the service sends the response, the port number identifies the virtual server that should receive the response.

The Use Proxy Port option becomes relevant if the use source IP (USIP) option is enabled. For TCP-based service types, such as TCP, HTTP, and SSL, the option is enabled by default. For UDP-based service types, such as UDP and DNS, including ANY, the option is disabled by default. For more information about the USIP option, see Enabling Use Source IP Mode.

To configure proxy port mode by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**useProxyPort** (**YES** | **NO**)

Example

```
set service Service-ANY-1 -useProxyPort YES
```

Parameters for configuring proxy port mode

ServiceName

The name of the service that you are configuring.

useProxyPort

If USIP is enabled, use a proxy port, instead of the source port in the client's request, as the source port when connecting to a physical server. Possible values: YES, NO. Default: YES for TCP based service types, NO for UDP based service types.

To configure proxy port mode by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to use the source IP address, and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Others**, in the **Use Proxy Port** dropdown list, select **YES.**
5. Click **OK**.

Setting a Limit on the Number of Client Connections

You can specify a maximum number of client connections that each load balanced server can handle. The NetScaler appliance then opens client connections to a server only until this limit is reached. When the load balanced server reaches its limit, monitor probes are skipped, and the server is not used for load balancing until it has finished processing existing connections and frees up capacity.

For more information on the Maximum Client setting, see the section Load Balancing Domain-Name Based Services.

Note: Connections that are in the process of closing are not considered for this limit.

To set a limit to the number of client connections by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**maxclient** <Value>

Example

set service Service-HTTP-1 -maxClient 1000

Parameters for configuring the maximum clients setting

ServiceName

The name of the service that you are configuring.

maxClient

Maximum number of open connections to the service. The default value is 0. The maximum value is 4294967294.

To set a limit to the number of client connections by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure the maximum number of client connections (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Thresholds**, in the **Max Clients** text box, type the maximum number of client connections (for example, **100**).
5. Click **OK**.

Setting a Limit on Number of Requests Per Connection to the Server

The NetScaler appliance can be configured to reuse connections to improve performance. In some scenarios, however, load balanced Web servers may have issues when connections are reused for too many requests. For HTTP or SSL services, use the max request option to limit the number of requests sent through a single connection to a load balanced Web server.

Note: You can configure the max request option for HTTP or SSL services only.

To limit the number of client requests per connection by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**maxReq** <Value>

Example

set service Service-HTTP-1 -maxReq 100

Parameters for configuring the maximum requests setting

ServiceName

The name of the service that you are configuring.

maxReq

Maximum number of requests that can be sent on a persistent connection to the service. The default value is 0. The minimum value is 0 and maximum value is 65535. ‘0’ specifies that there is no limit on the maximum requests that are sent on a persistent connection.

To limit the number of client requests per connection by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure the maximum number of client requests (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Thresholds**, in the **Max Requests** text box, type the maximum number of client requests (for example, **100**).
5. Click **OK**.

Setting a Threshold Value for the Monitors Bound to a Service

The NetScaler appliance designates a service as UP only when the sum of the weights of all monitors bound to it and that are UP is equal to or greater than the threshold value configured on the service. The weight for a monitor specifies how much that monitor contributes to designating the service to which it is bound as UP.

For example, assume that three monitors, named Monitor-HTTP-1, Monitor-HTTP-2, and Monitor-HTTP-3 respectively, are bound to Service-HTTP-1, and that the threshold configured on the service is three. Suppose the following weights are assigned to each monitor:

- The weight of Monitor-HTTP-1 is 1.
- The weight of Monitor-HTTP-2 is 3.
- The weight of Monitor-HTTP-3 is 1.

The service is marked UP only when one of the following is true:

- Monitor-HTTP-2 is UP.
- Monitor-HTTP-2 and Monitor-HTTP-1 or Monitor-HTTP-3 are UP
- All three monitors are UP.

To set the monitor threshold value on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**monThreshold** <Value>

Example

set service Service-HTTP-1 -monThreshold 100

Parameters for configuring the monitor threshold on a service

ServiceName

The name of the service that you are configuring.

monThreshold

Monitoring threshold. The default value is 0. The minimum value is 0 and maximum value is 65535.

To set the monitor threshold value on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure monitor threshold (for example, Service-HTTP-1), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. In the **Monitor Threshold** text box, type the monitor threshold (for example, **100**).
5. Click **OK**.

Setting a Timeout Value for Idle Client Connections

You can configure the service with a time-out value to terminate any idle client connections when the configured time elapses. If the client is idle during the configured time, the NetScaler closes the client connection.

To set a timeout value for idle client connections by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**cltTimeout** <Value>

Example

set service Service-HTTP-1 -cltTimeout 100

Parameters for setting a timeout value for idle client connections

ServiceName

The name of the service that you are configuring.

cltTimeout

Idle time (in seconds) after which the client connection is terminated. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

To set a timeout value for idle client connections by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure the time-out value for client connections (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Idle Time-out (secs)**, in the **Client** text box, type the timeout value (for example, **100**).
5. Click **OK**.

Setting a Timeout Value for Idle Server Connections

You can configure a service with a timeout value to terminate any idle server connections when the configured time elapses. If the server is idle for the configured amount of time, the NetScaler appliance closes the server connection.

To set a timeout value for idle server connections by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**svrTimeout** <Value>

Example

set service Service-HTTP-1 -svrTimeout 100

Parameters for configuring idle server timeout on services

ServiceName

The name of the service that you are configuring.

svrTimeout

Idle time (in seconds) after which the server connection is terminated. The default value is 360. The maximum value is 31536000.

To set a timeout value for idle server connections by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure the timeout value for server connections (for example, **Service-HTTP-1**), and click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Idle Time-out (secs)**, in the **Server** text box, type a timeout value as a number of seconds (for example, **100**).
5. Click **OK**.

Setting a Limit on the Bandwidth Usage by Clients

In some cases, servers may have limited bandwidth to handle client requests and may become overloaded. To prevent overloading a server, you can specify a maximum limit on the bandwidth processed by the server. The NetScaler appliance forwards requests to a load balanced server only until this limit is reached.

To set a maximum bandwidth limit on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**maxBandwidth** <Value>

Example

set service Service-HTTP-1 -maxBandwidth 100

Parameters for configuring a maximum bandwidth on a service

ServiceName

The name of the service that you are configuring.

maxBandwidth

Maximum bandwidth, in Kbps, allowed for forwarding incoming requests to a service. Possible Values: 0-limit of memory. Default: limit of memory.

To set set a maximum bandwidth limit on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details page, select the service for which you want to configure maximum bandwidth usage (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Under **Thresholds**, in the **Max Bandwidth (kbits)** text box, type the maximum bandwidth (for example, **100**).
5. Click **OK**.

Redirecting Client Requests to a Cache

You can configure a service to redirect client requests to a cache, and forward only those requests that are cache misses to a service chosen by the configured load balancing method.

To set cache redirection on a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**cacheable** <Value>

Example

set service Service-HTTP-1 -cacheable YES

Parameters for configuring caching of client requests

ServiceName

The name of the service that you are configuring.

cacheable

Access to disabled or DOWN services. If this option is enabled, and the service goes DOWN, all packets to the service are bridged. If this option is disabled, and the service goes DOWN, the packets are dropped. Possible values: YES and NO. Default: NO.

Note: For more information on cache redirection, see Cache Redirection.

To set cache redirection on a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to configure cache redirection (for example, **Service-HTTP-1**), and then click **Open**.
3. In the **Configure Service** dialog box, click the **Advanced** tab.
4. Scroll down, and under **Cache Redirection Options**, in **Cache Type** list, select the type of cache (for example, **Regular Server**).
5. Click **OK**.

Monitors

To manage a high-traffic load balancing setup, the NetScaler appliance needs to track the state of each load balanced server in near real time, so that it can divert traffic from any load balanced server that is not responding and send that traffic to a load balanced server that is responding. Therefore, a monitor is bound to each service. The monitor is configured to test the service by sending periodic probes to the service. (This is sometimes referred to as performing a health check.) If the monitor receives a timely response to its probes, it marks the service as UP. If it does not receive a timely response to the designated number of probes, it marks the service as DOWN.

The load balancing virtual server does not route requests to services that are DOWN. Those services are removed from its list of available services until they become available again and respond to monitor probes.

You can bind a single monitor or multiple monitors to the same service. If you bind multiple monitors to a service, they each evaluate responses to different types of traffic.

The NetScaler appliance supports built-in monitors to monitor common types of services. It also supports user-created monitors based on the built-in monitors, and allows you to create and configure custom monitors.

The Built-in Monitors

The NetScaler appliance contains a number of built-in monitors that you can use to monitor your services. These built-in monitors handle most of the common protocols. You cannot modify or remove the built-in monitors; you can only bind a built-in monitor to a service and unbind it from the service.

Note: You can create a custom monitor based on a built-in monitor. To learn how to create custom monitors, see Configuring Monitors in a Load Balancing Setup.

Monitoring TCP-based Applications

The NetScaler appliance has two built-in monitors that monitor TCP-based applications: **tcp-default** and **ping-default**. When you create a service, the appropriate default monitor is bound to it automatically, so that the service can be used immediately if it is UP. The tcp-default monitor is bound to all TCP services; the ping-default monitor is bound to all non-TCP services.

You cannot delete or modify default monitors. When you bind any other monitor to a TCP service, the default monitor is unbound from the service. The following table lists the monitor types, and the parameters and monitoring processes associated with each type.

Monitor type	Specific parameters	Process
tcp	Not applicable	The NetScaler appliance establishes a 3-way handshake with the monitor destination, and then closes the connection. If the appliance observes TCP traffic to the destination, it does not send TCP monitoring requests. This occurs if LRTM is disabled. By default, LRTM is disabled on this monitor.
http	httprequest [“HEAD /”] - HTTP request that is sent to the service. respcode [200] - A set of HTTP response codes are expected from the service.	The NetScaler appliance establishes a 3-way handshake with the monitor destination. After the connection is established, the appliance sends HTTP requests, and then compares the response code with the configured set of response codes.

tcp-ecv	send [""] - is the data that is sent to the service. The maximum permissible length of the string is 512 K bytes. recv [""] - expected response from the service. The maximum permissible length of the string is 128 K bytes.	The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send specific data to the service and expects a specific response through the receive parameter.
http-ecv	send [""] - HTTP data that is sent to the service recv [""] - the expected HTTP response data from the service	The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send the HTTP data to the service and expects the HTTP response that the receive parameter specifies. (HTTP body part without including HTTP headers). Empty response data matches any response. Expected data may be anywhere in the first 24K bytes of the HTTP body of the response.

udp-ecv	send [""] - data that is sent to the service. recv [""] - expected response from the service.	**When the receive string is specified:** If the response matches the receive string, the service is marked UP. If the response matches the receive string for a reverse monitor, the service is marked DOWN. The service is also marked as down if an “icmp port unreachable” message is received. **When the receive string is not specified:** The service is marked UP whether or not a response is received. The service is marked DOWN if an “icmp port unreachable” message is received. For LRTM monitors, when no response is received, the response time is the response time-out for the monitor. When the UDP monitors detect an ICMP port unreachable error, the service is marked DOWN immediately.
ping	Not Applicable	The NetScaler appliance sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.

To configure built-in monitors for TCP-based applications, see Configuring Monitors in a Load Balancing Setup.

Monitoring SSL Services

The NetScaler appliance has built-in secure monitors, TCPS and HTTPS. You can use the secure monitors to monitor HTTP as well as non-HTTP traffic. The secure monitors work as described below:

- **TCPS.** The NetScaler appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. After the handshake is over, the appliance closes the connection.
- **HTTPS.** The NetScaler appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. When the SSL connection is established, the appliance sends HTTP requests over the encrypted channel and checks the response codes.

The following table describes the available built-in monitors for monitoring SSL services.

Monitor type	Probe	Success criteria (Direct condition)
TCP	TCP connection SSL handshake	Successful TCP connection established and successful SSL handshake.
HTTP	TCP connection SSL handshake Encrypted HTTP request	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP response code in server HTTP response is encrypted.
TCP-ECV	TCP connection SSL handshake (Data sent to a server is encrypted.)	Successful TCP connection is established, successful SSL handshake is performed, and expected TCP data is received from the server.
HTTP-ECV	TCP connection SSL handshake (Encrypted HTTP request)	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP data is received from the server.

To configure built-in monitors to check the state of SSL services, see Configuring Monitors in a Load Balancing Setup.

Monitoring FTP Services

To monitor FTP services, the NetScaler appliance opens two connections to the FTP server. It first connects to the control port, which is used to transfer commands between a client and an FTP server. After it receives the expected response, it connects to the data port, which is used to transfer files between a client and an FTP server. Only when the FTP server responds as expected on both connections is it marked UP.

The NetScaler appliance has two built-in monitors for FTP services: the **FTP** monitor and the **FTP-EXTENDED** monitor. The FTP monitor checks basic functionality; the FTP-EXTENDED monitor also verifies that the FTP server is able to transmit a file correctly.

Parameter	Specifies
userName	User name used in the probe. Applies to both the FTP and FTP-EXTENDED monitor.
password	Password used in monitoring. Applies to both the FTP and FTP-EXTENDED monitor
fileName	File name to be used for FTP-EXTENDED monitor only.

To configure built-in monitors to check the state of FTP services, see Configuring Monitors in a Load Balancing Setup.

Monitoring SIP Services

The Session Initiation Protocol (SIP) is designed to initiate, manage, and terminate multimedia communications sessions. It has emerged as the standard for Internet telephony (VoIP). SIP messages can be transmitted over TCP or UDP. SIP messages are of two types: request messages and response messages.

The following table summarizes the structure of SIP messages.

Message type	Components	Components
Request	Method	Invite, Ack, Options, Bye, Cancel, Register
	Request URI	Represents the subject, media type, or urgency of sessions initiated. The common format is: sip:user:password@host:port;uri-parameters?headers
	SIP version	The SIP version being used
Response	SIP version	The SIP version that is being used.
	Status code	A 3-digit integer result code. The possible values are: 1xx: Information Responses. For example: 180, Ringing 2xx: Successful Responses. For example: 200, OK 3xx: Redirection Responses. For example: 302, Moved Temporarily 4xx: Request Failures Responses. For example: 403, Forbidden 5xx: Server Failure Responses. For example: 504, Gateway Time-out 6xx: Global Failure Responses. For example: 600, Busy Everywhere
	Reason-phrase	Textual description of the status code.

The traffic in an SIP-based communication system is routed through dedicated devices and applications (entities). In a multimedia communication session, these entities exchange messages.

One of the most common uses for SIP is VoIP, where SIP is used to set up the session. The following diagram illustrates how the messages and entities in a SIP-based communication system interoperate.

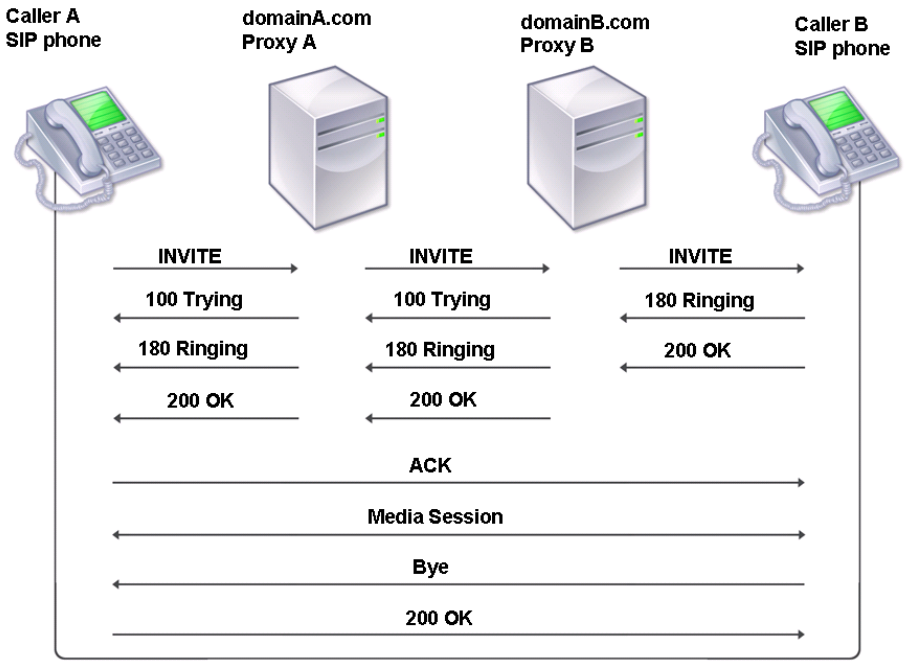


Figure 1. How SIP Works

The entity that initiates the call is referred to as the user agent (UA). The UA can be an SIP softphone (a PC-based application) or a SIP phone.

To initiate a call, the user agent sends an INVITE request to the previously configured SIP proxy server. The INVITE request contains the details of the destination, such as the destination uniform resource identifier (URI) and Call ID. In the diagram, the Caller A (user agent) sends an INVITE request to Proxy A.

When the proxy server receives the INVITE request, it sends a 100 (Trying) response to the user agent, Caller A. It also performs a DNS lookup to locate the SIP proxy server of the destination domain. After the SIP proxy server of the destination domain is located, the SIP proxy at the source domain sends the INVITE request to it. Here, Proxy A sends a 100 (Trying) response to Caller A and an INVITE request to Proxy B.

When the SIP proxy server of the destination domain receives the INVITE request from the SIP proxy server of the source domain, it responds with a 100 (Trying) response. It then sends the INVITE request to the destination user agent. In this case, Proxy B sends a 100 (Trying) response to Proxy A and an INVITE request to Caller B.

When the destination user agent receives the INVITE request, it alerts Caller B and responds with a 180 (ringing) response. This response is routed back to the source user agent through the proxies.

When caller B accepts the call, the destination user agent responds with a 200 (OK) response. This signifies that caller B has answered the call. This response is routed back to the source user agent through the proxies. After the call is set up, the user agents communicate directly without the proxies.

The following table describes the entities of a SIP-based communication system and their roles.

Entity	Role
User Agent (UA)	SIP user agents generate requests and respond to incoming requests. A user agent that generates requests is known as a User Agent Client (UAC). The user agent that responds to requests is known as the User Agent Server (UAS). In the preceding example, Caller A was the UAC and Caller B was the UAS.
Proxy Server	Proxies receive and route SIP requests based on the URI. They can selectively rewrite parts of the request message before forwarding it. They also handle registrations and invitations to user agents, and apply call policies.
Redirect Server	Redirect servers send routing information to the SIP proxy servers.
Registrar Server	Registrar servers provide location information to user agents and proxy servers.
Back-to-Back User Agent (B2BUA)	Back-to-Back User Agents (B2BUA) are combination of UAS and UAC.

You can configure the NetScaler appliance to load balance SIP requests to a group of SIP proxy servers. To do so, you need to create a load balancing virtual server with the load balancing method set to Call-ID hash, and then bind to it the services that are bound to the SIP proxies.

For load balancing to work, you must also configure the SIP proxies so that they do not add private IP addresses or private domains to the SIP header/payload. SIP proxies must add to the SIP header a domain name that resolves to the IP address of the SIP virtual server. Also, the SIP proxies must communicate with a common database to share registration information.

The NetScaler appliance can load balance SIP proxies in either a one-arm DSR configuration or an inline direct server return (DSR) configuration. In a one-arm DSR configuration, the appliance receives SIP requests from user agents and routes the requests to the appropriate SIP proxy by using the configured load balancing method. The SIP proxies send their responses to the destination SIP proxies, bypassing the appliance, as illustrated in the following diagram.

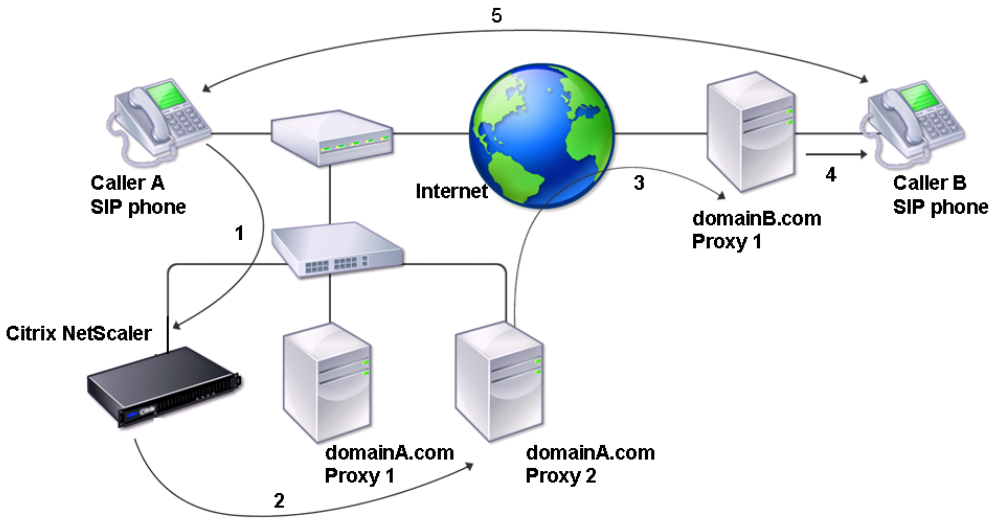


Figure 2. SIP in One-Arm Mode

The flow of requests and responses in this configuration is as follows:

- The user agent, Caller A, sends an INVITE request to the NetScaler. The NetScaler, using a load balancing method, routes the request to Proxy 2.
- Proxy 2 receives the INVITE request from the NetScaler and responds with a 100 (Trying) message.
- Proxy 2 performs a DNS lookup to obtain the IP address of the destination SIP proxy at domainB.com. It then sends the INVITE request to the destination proxy.
- The destination proxy responds with a 100 (Trying) message and sends the INVITE request to the destination user agent, Caller B. The destination user agent, Caller B, begins to ring and responds with a 180 (Ringing) message. This message is sent to Caller A through the NetScaler and the Proxy 2. After the user accepts the call, Caller B responds with a 200 (OK) message that is propagated to Caller A through the NetScaler and the Proxy 2.
- After Caller B accepts the call, the user agents (Caller A and Caller B) communicate independently.

In an inline DSR configuration, the appliance is placed between the router and the SIP proxy, as illustrated in the following diagram.

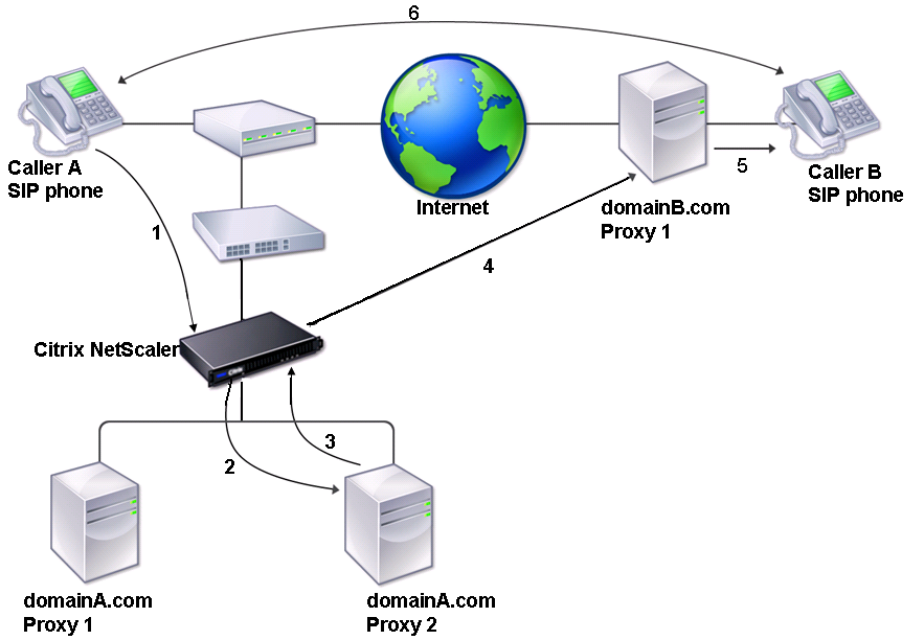


Figure 3. SIP in Inline Mode

The flow of requests and responses is as follows:

- The user agent, Caller A, sends an INVITE request to the appliance. The NetScaler, using a load balancing method, routes the request to Proxy 2.
- Proxy 2 receives the INVITE request from the appliance and responds with a 100 (Trying) message.
- Proxy 2 performs a DNS lookup to obtain the IP address of the destination SIP proxy at domainb.com. It then propagates the INVITE request to the destination proxy through the appliance.
- The appliance performs RNAT, and replaces the source IP address in the INVITE request with the NAT IP address, and then forwards the INVITE request to the destination SIP proxy.
- The destination proxy responds with a 100 (Trying) message and sends the INVITE request to the destination user agent, Caller B. Caller B begins to ring and responds with a 180 (Ringing) message. This message is sent to Caller A through the NetScaler and the Proxy 2. After the user accepts the call, Caller B responds with a 200 (OK) message that is propagated to Caller A through the appliance and Proxy 2.
- After the user accepts the call, the user agents (Caller A and Caller B) communicate independently.

Parameter	Specifies

maxForwards	SIP packet max-forwards. Possible Values: 0-255. Default: 1.
sipMethod	SIP method to be used for the query. Possible values: OPTIONS, INVITE, REGISTER Default value: OPTIONS.
sipURI	SIP method string, sent to the server. For example “OPTIONS sip:sip.test.”
sipregURI	SIP user to be registered.

To configure built-in monitors to check the state of SIP server, see Configuring Monitors in a Load Balancing Setup.

Monitoring RADIUS Services

The NetScaler appliance RADIUS monitor periodically checks the state of the RADIUS service to which it is bound by sending an authentication request to the service. The RADIUS server authenticates the RADIUS monitor and sends a response. By default, the monitor expects to receive a response code of 2, the default Access-Accept response, from the RADIUS server. As long as the monitor receives the appropriate response, it marks the service UP.

- If the client authenticated successfully, the RADIUS server sends an Access-Accept response. The default access-accept response code is 2, and this is the code that the appliance uses.
- If the client fails to authenticate successfully (such as when there is a mismatch in the user name, password, or secret key), the RADIUS server sends an Access-Reject response. The default access-reject response code is 3, and this is the code that the appliance uses.

Parameter	Specifies
userName	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
radKey	Shared secret key value that the RADIUS server uses during client authentication.
radNASid	NAS-ID that is encapsulated in the payload when an access request is made.
radNASip	The IP address that is encapsulated in the payload when an access-request is made. When radNASip is not configured, the NetScaler sends the mapped IP address (MIP) to the RADIUS server as the NAS IP address.

To monitor a RADIUS service, you must configure the RADIUS server to which it is bound as follows:

1. Add the user name and password of the client that the monitor will use for authentication to the RADIUS authentication database.
2. Add the IP address and secret key of the client to the appropriate RADIUS database.
3. Add the IP addresses that the appliance uses to send RADIUS packets to the RADIUS database. If the NetScaler appliance has more than one mapped IP address, or if a subnet IP address (SNIP) is used, you must add the same secret key for all of the IP addresses.

Caution: If the IP address used by the appliance are not added to the RADIUS database, the RADIUS server will discard all packets.

To configure built-in monitors to check the state of RADIUS server, see Configuring Monitors in a Load Balancing Setup.

Monitoring DNS and DNS-TCP Services

The NetScaler appliance has two built-in monitors that can be used to monitor DNS services: **DNS** and **DNS-TCP**. When bound to a service, either monitor periodically checks the state of that DNS service by sending a DNS query to it. The query resolves to an IPv4 or IPv6 address. That IP address is then checked against the list of test IP addresses that you configure. The list can contain up to five IP addresses. If the resolved IP address matches at least one IP address on the list, the DNS service is marked as up. If the resolved IP does not match any IP addresses on the list, the DNS service is marked as down.

Parameter	Parameter
query	The DNS query (domain name) sent to the DNS service that is being monitored. Default value: “\007” If the DNS query succeeds, the service is marked as UP; otherwise, it is marked as DOWN. For a reverse monitor, if the DNS query succeeds, the service is marked as DOWN; otherwise, it is marked as UP. If no response is received, the service is marked as DOWN.
queryType	The type of DNS query that is sent. Possible values: Address, Zone.
IPAddress	List of IP addresses that are checked against the response to the DNS monitoring probe.
IPv6	Select this check box if the IP address uses IPv6 format.

To configure the built-in DNS or DNS-TCP monitors, see Configuring Monitors in a Load Balancing Setup.

Monitoring LDAP Services

The NetScaler appliance has one built-in monitor that can be used to monitor LDAP services: the **LDAP** monitor. It periodically checks the LDAP service to which it is bound by authenticating and sending a search query to it. If the search is successful, the service is marked UP. If the LDAP server does not locate the entry, a failure message is sent to the LDAP monitor, and the service is marked DOWN.

You configure the LDAP monitor to define the search that it should perform when sending a query. You can use the Base DN parameter to specify a location in the directory hierarchy where the LDAP server should start the test query. You can use the Attribute parameter to specify an attribute of the target entity.

Parameter	Specifies
baseDN	Base name for the LDAP monitor from where the LDAP search must start. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.
bindDN	BDN name for the LDAP monitor.
filter	Filter for the LDAP monitor.
password	Password used in monitoring LDAP servers.
attribute	Attribute for the LDAP monitor.

To configure the built-in LDAP monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring MySQL Services

The NetScaler appliance has one built-in monitor that can be used to monitor MySQL services: the **MySQL** monitor. It periodically checks the MySQL service to which it is bound by sending a search query to it. If the search is successful, the service is marked UP. If the MySQL server does not respond or the search fails, a failure message is sent to the MySQL monitor, and the service is marked DOWN.

Parameter	Specifies
database	Database that is used for the MySQL monitor.
sqlQuery	SQL query that is used for the MySQL monitor.

To configure built-in MySQL monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring SNMP Services

The NetScaler appliance has one built-in monitor that can be used to monitor SMNP services: the **SNMP** monitor. It periodically checks the SNMP agent on the service to which it is bound by sending a query for the enterprise identification ID (OID) that you configure for monitoring. If the query is successful, the service is marked UP. If the SNMP service finds the OID that you specified, the query succeeds and the SNMP monitor marks the service UP. If it does not find the OID, the query fails and the SNMP monitor marks service DOWN.

Parameter	Specifies
SNMPOID	OID that is used for the SNMP monitor.
snmpCommunity	Community that is used for the SNMP monitor.
snmpThreshold	Threshold that is used for the SNMP monitor.
snmpVersion	SNMP version that is used for load monitoring. Possible Values: V1, V2.

To configure the built-in SNMP monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring NNTP Services

The NetScaler appliance has one built-in monitor that can be used to monitor NNTP services: the **NNTP** monitor. It periodically checks the NNTP service to which it is bound by connecting to the service and checking for the existence of the newsgroup that you specify. If the newsgroup exists, the search is successful and the service is marked UP. If the NNTP service does not respond or the search fails, the service is marked DOWN.

The NNTP monitor can optionally be configured to post a test message to the newsgroup as well.

Parameter	Specifies
userName	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
group	Group name to be queried for NNTP monitor.

To configure the built-in NNTP monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring POP3 Services

The NetScaler appliance has one built-in monitor that can be used to monitor POP3 services: the POP3 monitor. It periodically checks the POP3 service to which it is bound by opening a connection with a POP3 server. If the POP3 server responds with the correct response codes within the configured time period, it marks the service UP. If the POP3 service does not respond, or responds incorrectly, it marks the service DOWN.

Parameter	Specifies
userName	User name POP3 server. This user name is used in the probe.
password	Password used in monitoring POP3 servers.
scriptName	The path and name of the script to execute.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in POP3 monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring SMTP Services

The NetScaler appliance has one built-in monitor that can be used to monitor SMTP services: the **SMTP** monitor. It periodically checks the SMTP service to which it is bound by opening a connection with it and conducting a series of handshakes to ensure that the server is operating correctly. If the SMTP service completes the handshakes properly, the monitor marks the service UP. If the SMTP service does not respond, or responds incorrectly, it marks the service DOWN.

Parameter	Specifies
userName	User name SMTP server. This user name is used in the probe.
password	Password used in monitoring SMTP servers.
scriptName	The path and name of the script to execute.
dispatcherIP	The IP Address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in SMTP monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring RTSP Servers

The NetScaler appliance has one built-in monitor that can be used to monitor RTSP services: the RTSP monitor. It periodically checks the RTSP service to which it is bound by opening a connection with the load balanced RTSP server. The type of connection that it opens, and the response that it expects, differs depending upon the network configuration. If the RTSP service responds as expected within the configured time period, it marks the service UP. If the service does not respond, or responds incorrectly, it marks the service DOWN.

The NetScaler appliance can be configured to load balance RTSP servers using two topologies: NAT-off and NAT-on. RTSP servers send their responses directly to the client, bypassing the appliance. The appliance must be configured to monitor RTSP services differently depending upon which topology your network uses. The appliance can be deployed either in inline or non-inline mode in both NAT-off and NAT-on mode.

In NAT-off mode, the appliance operates as a router: it receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. If your load balanced RTSP servers are assigned publicly accessible FQDNs in DNS, the load balanced servers send their responses directly to the client, bypassing the appliance. The following figure demonstrates this configuration.

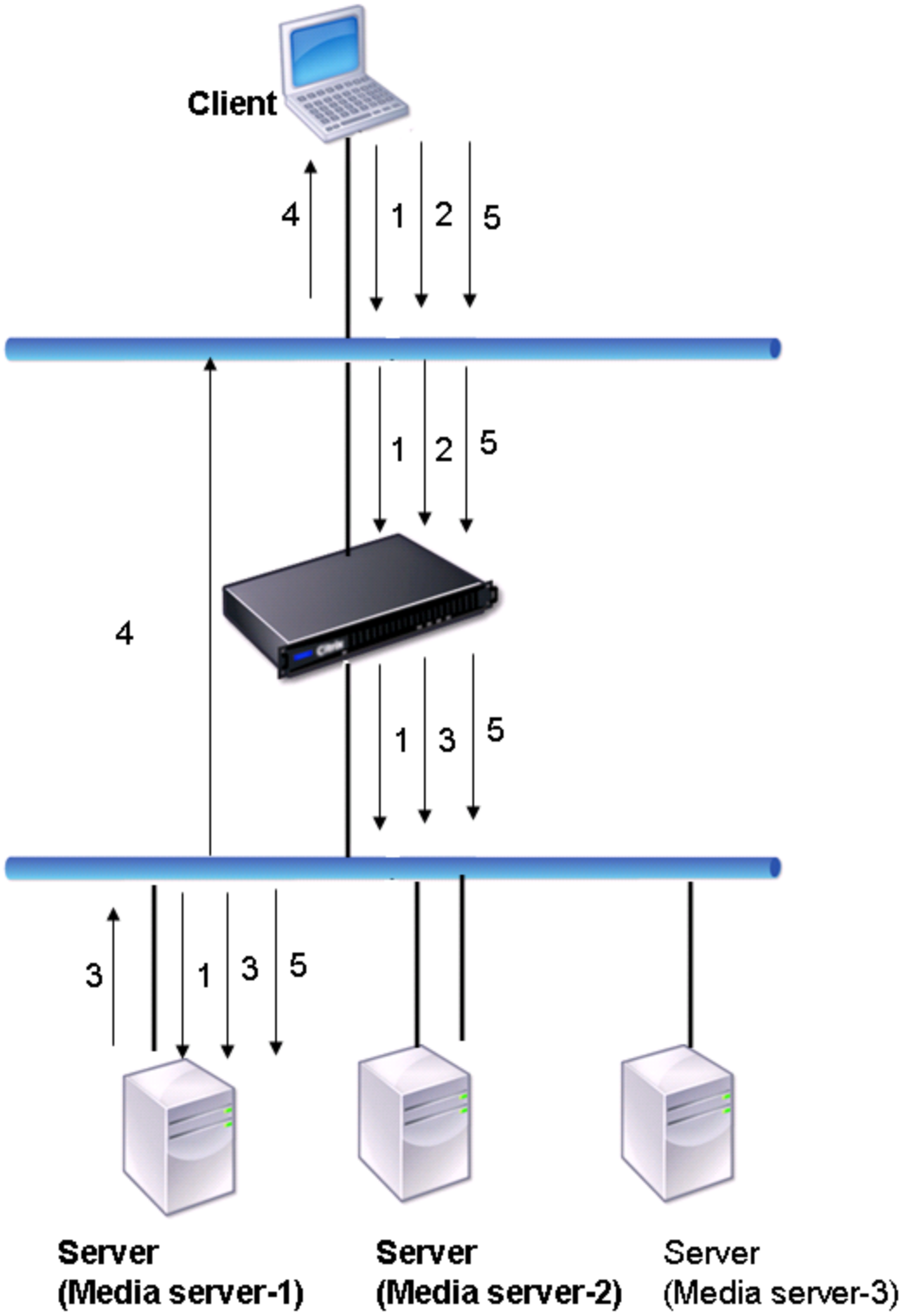


Figure 1. RTSP in NAT-off Mode

The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media

Server-1.

2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:

   - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
   - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request may be sent to a different service.

3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.

   Note: The appliance does not perform NAT to identify the RTSP connection, because the RTSP connections bypass it.

4. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the media server. Media Server-1 performs the requested actions, such as play, forward, or rewind.

In NAT-on mode, the appliance receives RTSP requests from the client and routes those requests to the appropriate media server using the configured load balancing method. The media server then sends its responses to the client through the appliance, as illustrated in the following diagram.

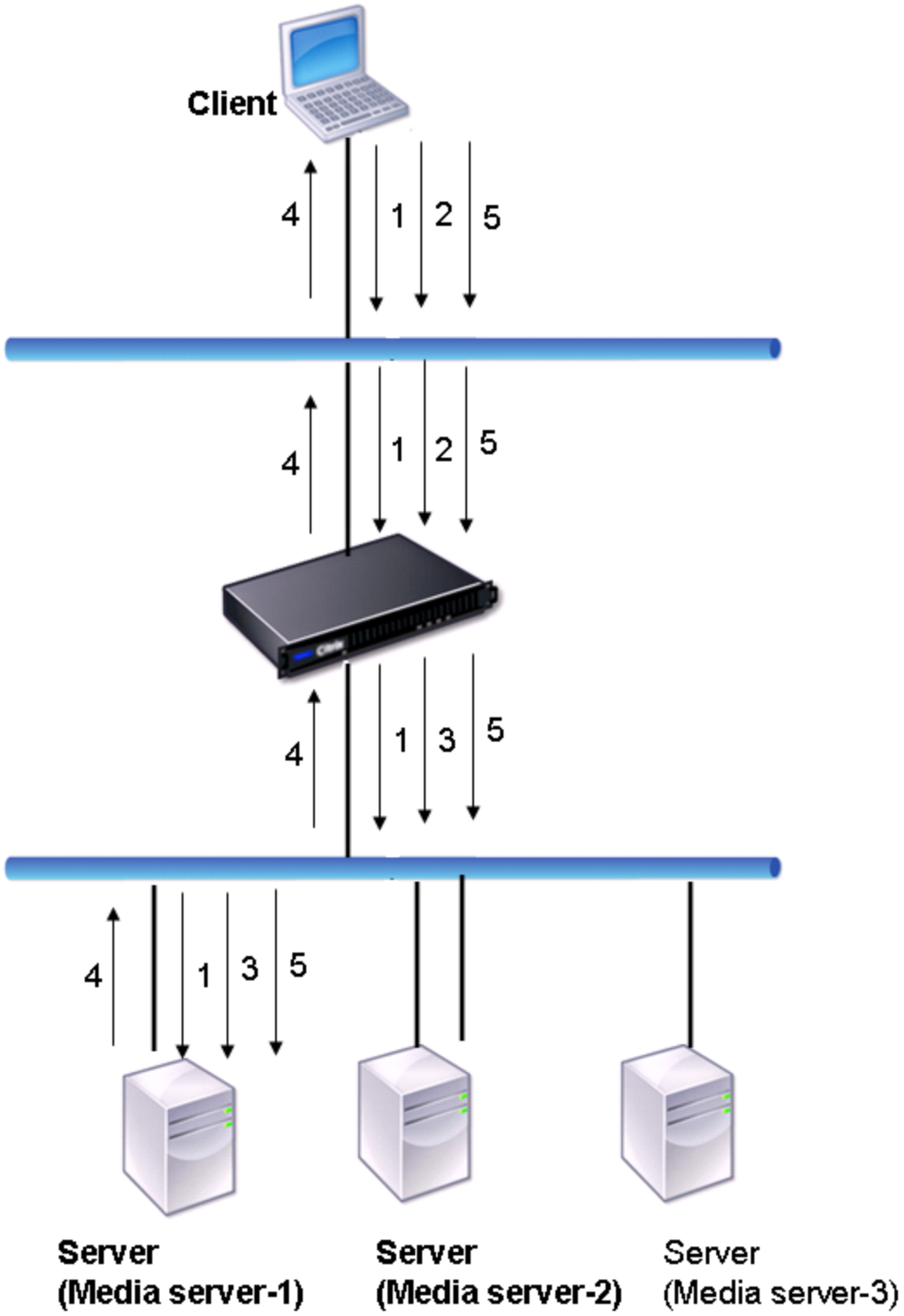


Figure 2. RTSP in NAT-on Mode

The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media

Server-1.

2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using the RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:

   - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
   - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request may be sent to a different service.

3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.
4. The appliance performs NAT to identify the client for RTSP data connections, and the RTSP connections pass through the appliance and are routed to the correct client.
5. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the appliance. The appliance uses RTSPSID persistence to identify the appropriate service, and routes the request to Media Server-1. Media Server-1 performs the requested action, such as play, forward, or rewind.

The RTSP monitor uses the RTSP protocol to evaluate the state of the RTSP services. The RTSP monitor connects to the RTSP server and conducts a sequence of handshakes to ensure that the server is operating correctly.

Parameter	Specifies
rtspRequest	The RTSP request string that is sent to the RTSP server (for example, OPTIONS *). The default value is 07. The length of the request must not exceed 163 characters.
respCode	Set of response codes that are expected from the service.

For instructions on configuring an RTSP monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring the XML Broker Services

The NetScaler appliance has a built-in monitor type, **CITRIX-XML-SERVICE**, with which you can create monitors to monitor the XML Broker services. The XML Broker services are used by Citrix® XenApp™. The monitor opens a connection to the service and periodically probes the XML services to which it is bound. If the server responds as expected within the configured time period, the monitor marks the service UP. If the service does not respond, or responds incorrectly, the monitor marks the service DOWN.

To configure a CITRIX-XML-SERVICE monitor, you need to specify the application name in addition to setting the standard parameters. The application name is the name of the application that has to be run to monitor the state of the XML Broker service. The default application is Notepad.

To configure monitors for XML Broker services, see Configuring Monitors in a Load Balancing Setup.

Monitoring ARP Requests

The NetScaler appliance has one built-in monitor that can be used to monitor ARP requests: the **ARP** monitor. This monitor periodically sends an ARP request to the service to which it is bound, and listens for the expected response. If it receives the expected response, it marks the service UP. If it receives no response or the wrong response, it marks the service DOWN.

ARP locates a hardware address for a load balanced server when only the network layer address is known. ARP works with IPv4 to translate IP addresses to Ethernet MAC addresses. ARP monitoring is not relevant to IPv6 networks, and is therefore not supported on those networks.

There are no special parameters for the ARP monitor.

For instructions on configuring an ARP monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring the Access Gateway

The NetScaler appliance has one built-in monitor that can be used to monitor a load-balanced Citrix Access Gateway: the **CITRIX-AG** monitor. This is in addition to two monitors for the Advanced Access Control login page and agent service page, which are described separately. The CITRIX-AG monitor periodically logs on to the Access Gateway service to which it is bound, and awaits the expected responses to its requests. If it receives the expected responses, it marks the service UP. If it receives no response or the wrong responses, it marks the service DOWN.

To configure monitoring of an Access Gateway, you must first create a local user and password for the monitor on the load balanced Access Gateway server that the service is bound to. After you configure the Access Gateway, you then configure the monitor. The monitor logs on to the Access Gateway using the realm and user name. For example, if you configured a realm of LDAP and a user name of user1, the Access Gateway logs on as LDAP/user1.

Note: RSA SecurID authentication is not supported for this monitor. RSA SecurID requires an RSA-generated token as a password, which is not supported on the NetScaler appliance.

Parameter	Specifies
userName	A user name.
password	A password for the username.
secondaryPassword	A secondary password for the username.

For instructions on configuring the CITRIX-AG monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring the Advanced Access Control Login Page

The NetScaler appliance has one built-in monitor that can be used to monitor the Advanced Access Control (AAC) login page on a load-balanced Citrix Access Gateway: the **CITRIX-AAC-LOGINPAGE** monitor. This monitor periodically logs on to the AAC login page via the Access Gateway service to which it is bound, and awaits the expected response. If it receives the expected response, it marks the service UP. If it receives no response or the wrong response, it marks the service DOWN.

Parameter	Specifies
logonpointName	The URL from which users access corporate resources using the Access Gateway Advanced edition. This setting controls access to server farms, the Access Interface configuration, and other session-specific settings. It can also e used as a filter within Access Gateway policies.

For instructions on configuring the CITRIX-AAC-LOGINPAGE monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring the Advanced Access Control Logon Agent Service Page

The NetScaler appliance has one built-in monitor that can be used to monitor the Advanced Access Control (AAC) agent service page on a load-balanced Citrix Access Gateway: the **CITRIX-AAC-LAS** monitor. The Logon Agent Service (LAS) is a service component of Advanced Access Control that requests authentication to the Authentication Service. This monitor periodically logs on to the AAC agent service page via the Access Gateway service to which it is bound, and awaits the expected response. If it receives the expected response, it marks the service UP. If it receives no response or the wrong response, it marks the service DOWN.

Parameter	Specifies
logonpointName	The URL from which users access corporate resources using the Access Gateway Advanced edition. This setting controls access to server farms, the Access Interface configuration, and other session-specific settings. It can also e used as a filter within Access Gateway policies.
lasVersion	The version number of the agent.

For instructions on configuring the CITRIX-AAC-LAS monitor, see Configuring Monitors in a Load Balancing Setup.

Monitoring the Dynamic Desktop Controller (DDC) Services

In desktop virtualization, the NetScaler appliance can be used to load balance the Web Interface (WI) servers and the Dynamic Desktop Controller (DDC) servers deployed by Citrix® XenDesktopTM environment. The NetScaler provides a built-in monitor, **CITRIX-XD-DDC** monitor, which monitors the DDC servers. In addition to the health check, you can also verify whether the probe is sent by a valid user of the DDC server.

The monitor sends a probe to the DDC server in the form of an XML message. If the DDC server responds to the probe with the identity of the server farm, the probe is considered to be successful and the server's status is marked as UP. If the HTTP response does not have a success code or the identity of the server farm is not present in the response, the probe is considered to be a failure and the server's status is marked as DOWN.

The **Validate Credentials** option determines the probe to be sent by the monitor to the DDC server, that is, whether to request only the server name or to also validate the login credentials.

Note: Regardless of whether or not the user credentials (user name, password and domain) are specified on the XD-DDC monitor, the DDC server validates the user credentials only if the option to validate credentials is enabled on the monitor.

If you use the wizard for configuring the load balancing of the XenDesktop servers, the XD-DDC monitor is automatically created and bound to the DDC services. If you do not use the wizard, add a monitor of the type XD-DDC.

- For instructions on using the wizard, see Configuring the load balancing of XenDesktop.
- For instructions on adding a monitor, see Creating Monitors.
- For instructions on binding a monitor to a service, see Binding Monitors to Services.

To add an XD-DDC monitor with the validate credentials option by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add an XD-DDC monitor and verify the configuration:

- add lb monitor <monitorName> <monitorType> **-userName** <userName> **-password** <password> **-ddcDomain** <ddc_domain_name> **-validateCred** YES
- show lb monitor <monitorName>

Example

```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -password E12Dc35450a1 -ddcDomain dh
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval.....:..5 sec...Retries.....:..3
Response timeout.....:..2 sec...Down time.....:..30 sec
Reverse.....:..NO...Transparent.....:..NO
Secure.....:..NO...LRTM.....:..ENABLED
Action.....:..Not applicable...Deviation.....:..0 sec
Destination IP.....:..Bound service
Destination port.....:..Bound service
Iptunnel.....:..NO
TOS.....:..NO...TOS ID.....:..0
SNMP Alert Retries.....:..0...Success Retries.....:..1
Failure Retries.....:..0

Special parameters:
User Name.....:"Administrator"
Password.....:*****
DDC Domain.....: "dhop"
Done
```

To specify the validate credentials option on an XD-DDC monitor by using the NetScaler command line

At the NetScaler command prompt, type:

set lb monitor <monitorName> <monitorType> **-userName -password -ddcDomain** <ddc_domain_name> **-validateCred** YES

Example

```
 > set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName Administrator -password D123S1R2A
Done
```

Parameters for configuring a monitor

monitorName

Name to identify the monitor.

monitorType

Type of the monitor. For monitoring DDC servers, specify CITRIX-XD-DDC.

userName

User name of the user account authorized to log into the DDC server.

password

Password for the user account.

ddcDomain

Domain in which the DDC server is present.

validateCred

Verify the validity of the user credentials. Possible values: YES, NO. Default: NO

To configure an XD-DDC monitor with the validate credentials option by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, do one of the following:
   - To add an XD-DDC monitor, click **Add**.
   - To modify an XD-DDC monitor, select the monitor, and click **Open**.
3. Type a name for the monitor.
4. Select the monitor type as **CITRIX-XD-DDC**.
5. On the **Special Parameters** tab, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a monitor" as shown:
   - **Validate Credentials**—validateCred (To specify YES, select the check box.)
   - **Name***—monitorName
   - **Type***—monitorType
   - **User Name***—userName
   - **Password***—password
   - **Domain Name***—ddcDomain

   *A required parameter
6. Click **Create**.
7. Select the new monitor, click **Open**, and verify the settings.

Monitoring Web Interface Services

In desktop virtualization, the NetScaler appliance can be used to load balance the Web Interface (WI) servers and Dynamic Desktop Controller (DDC) servers deployed in the Citrix® XenApp™ and Citrix® XenDesktop™ and environments. The NetScaler appliance has two built-in monitor types for monitoring the WI servers used in these environments.

A CITRIX-WEB-INTERFACE monitor can monitor the Web Interface services efficiently because it monitors a dynamic page at the location specified by the site path. The monitor checks for critical failures in resource availability.

When you configure a CITRIX-WEB-INTERFACE monitor, specify the site path to the location of the http page that displays the data collected by the monitor. To monitor the status of the service, in the specified site path, you can view the data updated dynamically by the monitoring script auth/nocookies.aspx.

Note: End the site path with a slash (/) to indicate that the monitored resource is dynamic.

Note: When you configure the WI-EXTENDED monitor, when specifying the site path, do not enter a slash (/) at the end of the path as the software internally adds a slash at the end of the path. For example, note the following command:

add monitor **wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb -domain ccc**

A CITRIX-WI-EXTENDED monitor verifies the logging process with the Web Interface service. This monitor accesses the login page and passes the user name, password, domain, and site path that were specified while configuring the monitor. It verifies the validity of the login credentials, correct configuration of the monitor (for example, the site path), and the connection with the IIS server.

Note: The CITRIX-WI-EXTENDED monitor is supported only for the .NET version of the WI servers. This monitor will not work for the JSP version of the WI servers.

If you use the wizard for configuring load balancing of the XenDesktop servers, a CITRIX-WEB-INTERFACE monitor is automatically created and bound to the WI services. The wizard adds and binds a CITRIX-WEB-INTERFACE monitor by default. If you want to add and bind a CITRIX-WI-EXTENDED monitor, select the **Validate Credentials** check box and type the necessary data. If you do not use the wizard, add a monitor corresponding to the WI services and bind it to each WI service that you create.

- For instructions on using the wizard, see Configuring XenDesktop for Load Balancing or Configuring XenApp for Load Balancing.
- For instructions on adding a CITRIX-WEB-INTERFACE monitor, see Creating Monitors.
- For instructions on binding a monitor to a service, see Binding Monitors to Services.

To add a WI monitor by using the NetScaler command line

At the NetScaler command prompt, type:

add lb monitor <monitorName> <monitorType> **-sitePath** <site_path> **-dispatcherIP 127.0.0.1 -dispatcherPort 3013 -userName** <username> **-password** <password> **-domain** <domain_name>

Examples

add lb monitor mwie CITRIX-WEB-INTERFACE -sitePath "/Citrix/XDWI/"

add lb monitor mwie CITRIX-WI-EXTENDED -sitePath "/Citrix/XDWI/" -dispatcherIP 127.0.0.1 -dispatcherPort 3013 -userName administrator -password d83d154575d426 -encrypted -domain wi

Parameters for configuring WI monitors

monitorName

Name of the monitor.

monitorType

Type of the monitor. Type of monitor. For monitoring WI servers, specify CITRIX-WEB-INTERFACE or CITRIX_WI_EXTENDED.

sitePath

URL of the logon page.

password

Password for the user account. To view the dynamic page, this site path must end with a slash (/).

userName

User name of the user account authorized to log on to the WI server.

password

Password for the user account.

domain

Domain in which the WI server is present.

To add a WI monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, do one of the following:
   - To add a WI monitor, click **Add**.
   - To modify a WI monitor, select the monitor, and click **Open**.
3. Type a name for the monitor.
4. Select the monitor type as **CITRIX-WEB-INTERFACE** or **CITRIX-WI-EXTENDED**.
5. On the **Special Parameters** tab, type the site path. To configure the CITRIX-WI-EXTENDED monitor, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a monitor" as shown:
   - **User Name*****—userName
   - **Password*****—password
   - **Domain Name*****—domain
6. Click **Create**.
7. Select the new monitor, click **Open**, and verify the settings.

Custom Monitors

In addition to built-in monitors, you can use custom monitors to check the state of your services. The NetScaler appliance provides several types of custom monitors based on scripts that are included with NetScaler operating system that can be used to determine the state of services based on the load on the service or network traffic sent to the service. These are the inline monitors, user monitors, and load monitors.

With any of these types of monitors, you can use the supplied functionality, or you can create your own scripts and use those scripts to determine the state of the service to which the monitor is bound.

Configuring Inline Monitors

Inline monitors analyze and probe the responses from the services to which they are bound only when those services receive client requests. The inline monitor is of type HTTP-INLINE and can only be configured to work with HTTP and HTTPS services. An inline monitor determines that the service to which it is bound is UP by checking its responses to the requests that are sent to it. When no client requests are sent to the service, the inline monitor probes the service by using the configured URL.

Note: Inline monitors cannot be bound to HTTP or HTTPS Global Server Load Balancing (GSLB) remote or local services because these services represent virtual servers rather than actual load balanced Web servers.

Inline monitors have a time-out value and a retry count when probes fail. You can select any of the following action types for the NetScaler appliance to take when a failure occurs:

- **NONE.** No explicit action is taken. You can view the service and monitor, and the monitor indicates the number of current contiguous error responses and cumulative responses checked.
- **LOG.** Logs the event in ns/syslog and displays the counters.
- **DOWN.** Marks the service down and does not direct any traffic to the service. This setting breaks any persistent connections to the service. This action also logs the event and displays counters.

After the service is down, the service remains DOWN for the configured down time. After the DOWN time elapses, the inline monitor uses the configured URL to probe the service to see if it is available again. If the probe succeeds, the state of the service is changed to UP. Traffic is directed to the service, and monitoring resumes as before.

To configure inline monitors, see Configuring Monitors in a Load Balancing Setup.

Understanding User Monitors

User monitors extend the scope of custom monitors. You can create user monitors to track the health of customized applications and protocols that the NetScaler appliance does not support. The following diagram illustrates how the user monitor works.

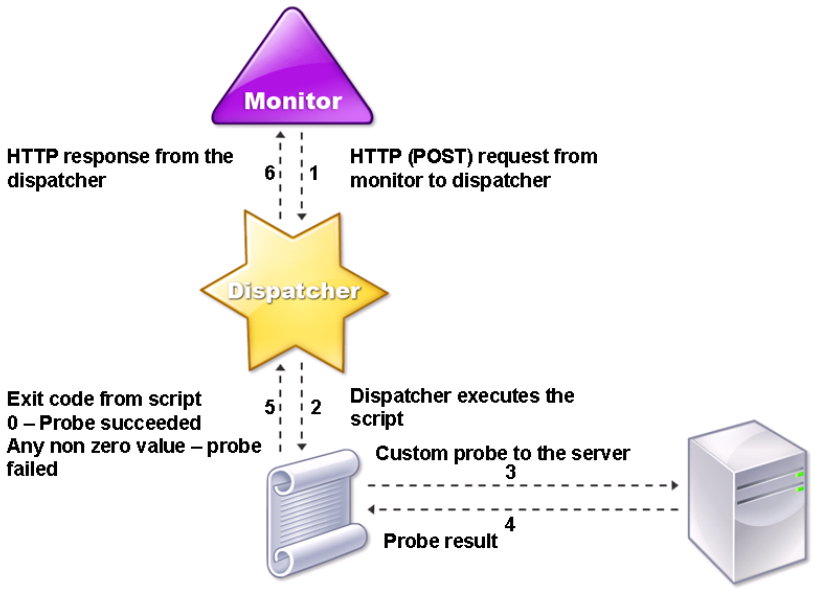


Figure 1. User Monitors

A user monitor requires the following components.

- **Dispatcher.** A process on the appliance that listens to monitoring requests. A dispatcher can be on the loopback IP address (127.0.0.1) and port 3013. Dispatchers are also known as internal dispatchers. A dispatcher can also be a Web server that supports Common Gateway Interface (CGI). Such dispatchers are also known as external dispatchers. They are used for custom scripts that do not run on the FreeBSD environment, such as .NET scripts.

  Note: You can configure the monitor and the dispatcher to use HTTPS instead of HTTP if you enable the “secure” option on the monitor and configure it as an external dispatcher. However, an internal dispatcher understands only HTTP, and cannot use HTTPS.

  In a HA setup, the dispatcher runs on both the primary and secondary NetScaler appliances. The dispatcher remains inactive on the secondary appliance.

- **Script.** The script is a program that sends custom probes to the load balanced server and returns the response code to the dispatcher. The script may return any value to the dispatcher, but if a probe succeeds, the script must return a value of zero (0). The dispatcher considers any other value as probe failure.

  The NetScaler appliance is bundled with sample scripts for commonly used protocols. The scripts exist in the /nsconfig/monitors directory. If you want to add a new script, you add the script there. If you want to customize an existing script, you copy the script with a new name and modify the script.

  For the scripts to function correctly, the name of the script file must not exceed 63 characters, and the maximum number of script arguments is 512. To debug the script, you must run it using the nsumon-debug.pl script from the NetScaler command line. You use the script name (with its arguments), IP address, and the port as the arguments of the nsumon-debug.pl script. Users must use the script name, IP address, port, time-out, and the script arguments for the nsumon-debug.pl script.

To track the status of the server, the monitor sends an HTTP POST request to the configured dispatcher. This POST request contains the IP address and port of the server, and the script that must be executed. The dispatcher executes the script as a child process, with user-defined parameters (if any). Then, the script sends a probe to the server. The script sends the status of the probe (response code) to the dispatcher. The dispatcher converts the response code to an HTTP response and sends it to the monitor. Based on the HTTP response, the monitor marks the service as up or down.

The appliance logs the error messages to the /var/nslog/nsumond.log file when user monitor probes fail. The following table lists the user monitors and the possible reasons for failure.

User monitor type	Probe failure reasons
SMTP	Monitor fails to establish a connection to the server.
NNTP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Monitor fails to find NNTP group.
LDAP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Monitor fails to bind to the LDAP server.
	Monitor fails to locate an entry for the target entity in the LDAP server.
FTP	The connection to the server times out.

	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Login fails.
	Monitor fails to find the file on the server.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Login fails.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Login fails.
	Preparation of SQL query fails.
	Execution of SQL query fails.
SNMP	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Login fails.
	Monitor fails to create SNMP session.
	Monitor fails to find the object identifier.
	The monitor threshold value setting is greater than or equal to the actual threshold of the monitor.
RDP (Windows Terminal Server)	Missing or invalid script arguments, which may include an invalid number of arguments or argument format.
	Monitor fails to create a socket.
	Mismatch in version.
	Monitor fails to confirm connection.

You can view the log file from the NetScaler command line by using the following commands, which open a BSD shell, display the log file on the screen, and then close the BSD shell and return you to the NetScaler command prompt:

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
```

User monitors also have a time-out value and a retry count on failure of probes. You can use user monitors with non-user monitors. During high CPU utilization, a non-user monitor enables faster detection of a server failure.

Note: If the user monitor probe times out during high CPU usage, the state of the service remains unchanged.

How to Use a User Monitor to Check Web Sites

You can configure a user monitor to check for specific Web site problems that are reported by HTTP servers using specific HTTP codes. The following table lists the HTTP response codes that this user monitor expects.

HTTP response code	Meaning
200 - success	Probe success.
503 - service unavailable	Probe failure.
404 - not found	Script not found or cannot execute.
500 - Internal server error	Internal error/resource constraints in dispatcher (out of memory, too many connections, unexpected system error, or too many processes). The service is not marked DOWN.
400 - bad request	Error parsing HTTP request.
502 - bad gateway	Error decoding script's response.

You configure the user monitor for HTTP by using the following parameters.

Parameter	Specifies
scriptName	The path and name of the script to execute.
scriptArgs	The strings that are added in the POST data. They are copied to the request verbatim.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.
localfileName	The name of a monitor script file on the local system.
destPath	A particular location on the NetScaler appliance where the uploaded local file is stored.

To create a user monitor to monitor HTTP, see Configuring Monitors in a Load Balancing Setup.

Understanding the Internal Dispatcher

You can use a custom user monitor with the internal dispatcher. Consider a case where you need to track the health of a server based on the presence of a file on the server. The following diagram illustrates this scenario.

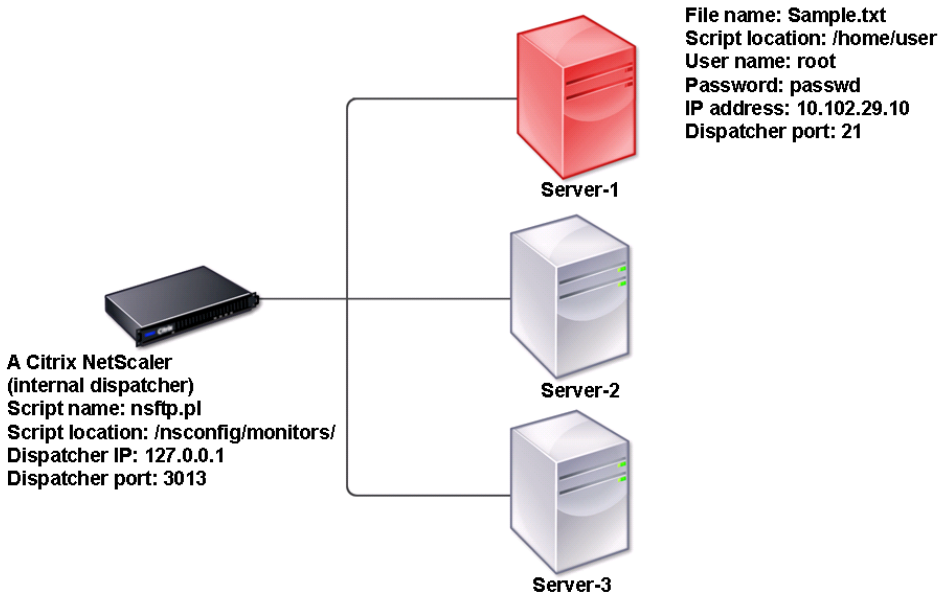
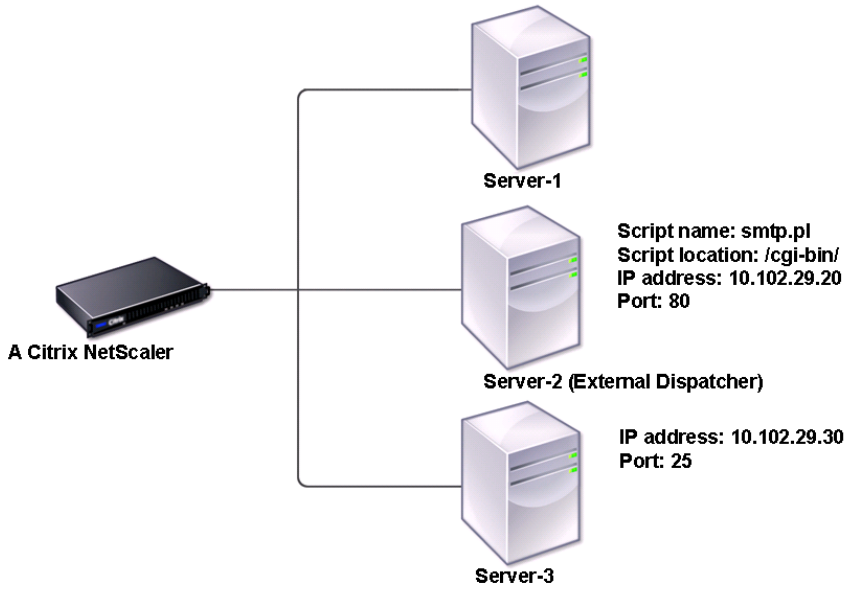


Figure 1. Using a User Monitor with the Internal Dispatcher

A possible solution is to use a Perl script that initiates an FTP session with the server and checks for the presence of the file. You can then create a user monitor that uses the Perl script. The NetScaler includes such a Perl script (nsftp.pl), in the /nsconfig/monitors/ directory.

You can use a user monitor with an external dispatcher. Consider a case where you must track the health of a server based on the state of an SMTP service on another server. This scenario is illustrated in the following diagram.

Figure 2. Using a User Monitor with an External Dispatcher



A possible solution would be to create a Perl script that checks the state of the SMTP service on the server. You can then create a user monitor that uses the Perl script.

Configuring a Custom User Monitor

To configure a custom user monitor, you must first write the script that performs the action that the monitor will use to check the service that is bound to it, and upload the script to the /home/user directory on the NetScaler appliance. Then you create the monitor on the appliance, as described below.

To configure a user monitor by using the NetScaler command line

At the NetScaler command prompt, type:

add monitor <MonitorName> USER -**scriptname** <NameOfScript> -**scriptargs** <Arguments>

Example

add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file=/home/user/sample.txt;user=root;password=passwd"

Understanding Load Monitors

Load monitors use SNMP polled OIDs to calculate load. The load monitor uses the IP address of the service to which it is bound (the destination IP address) for polling. It sends an SNMP query to the service, specifying the OID for a metric. The metrics can be CPU, memory, or number of server connections. The server responds to the query with a metric value. The metric value in the response is compared with the threshold value. The NetScaler appliance considers the service for load balancing only if the metric is less than the threshold value. The service with the lowest load value is considered first.

The following diagram illustrates a load monitor configured for the services described in the basic load balancing setup discussed in Setting Up Basic Load Balancing.

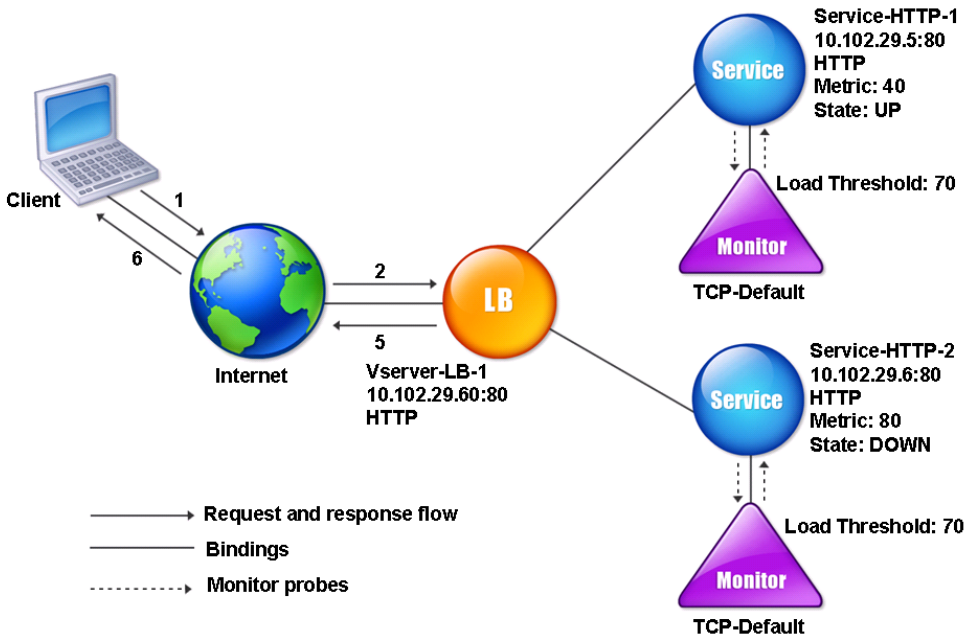


Figure 1. Operation of Load Monitors

Note: The load monitor does not determine the state of the service. It only enables the appliance to consider the service for load balancing.

After you configure the load monitor, you must then configure the metrics that the monitor will use. For load assessment, the load monitor considers server parameters known as metrics, which are defined within the metric tables in the appliance configuration. Metric tables can be of two types:

- **Local.** By default, this table exists in the appliance. It consists of four metrics: connections, packets, response time, and bandwidth. The appliance specifies these

metrics for a service, and SNMP queries are not originated for these services. These metrics cannot be changed.

- **Custom.** A user-defined table. Each metric is associated with an OID.

By default, the appliance generates the following tables:

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL
- FOUNDRY
- ALTEON

You can either add the appliance-generated metric tables, or you can add tables of your own choosing, as shown in the following table. The values in the metric table are provided only as examples. In an actual scenario, consider the real values for the metrics.

Metric name	OIDs	Weight	Threshold
CPU	1.2.3.4	2	70
Memory	4.5.6.7	3	80
Connections	5.6.7.8	4	90

To calculate the load for one or more metrics, you assign a weight to each metric. The default weight is 1. The weight represents the priority given to each metric. If the weight is high, the priority is high. The appliance chooses a service based on the SOURCEIPDESTIP hash algorithm.

You can also set the threshold value for each metric. The threshold value enables the appliance to select a service for load balancing if the metric value for the service is less than the threshold value. The threshold value also determines the load on each service.

Configuring Load Monitors

To configure a load monitor, first create the load monitor. For instructions on creating a monitor, see Creating Monitors. Next, select or create the metric table to define a set of metrics that determine the state of the server, and (if you create a metric table) bind each metric to the metric table.

To create a metric table by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add metricTable <metricTableName>
- bind metricTable <metricTableName> <metric> <SNMPOID>

Example

add metricTable Table-Custom-1

bind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11

Parameter for configuring load balancing metric tables

metricTableName

Name of the metric table. This alphanumeric string is required and cannot be changed after the metric table is created. The name must not exceed 31 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

metric

The name of the metric that you are binding to the metric table.

SNMPOID

The SNMP OID for the metric that you are binding to the metric table.

To create a metric table and bind metrics to it by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Metric Tables**.
2. In the details pane, click **Add**.
3. In the **Create Metric Table** dialog box, in the **Metric Table Name** text box, type the name of the metric table (for example, **Table-Custom-1**).
4. Click **Create**.
5. In the details pane, select the metric table that you just created (for example, **Table-Custom-1**), and then click **Open**.
6. In the **Configure Metric Table** dialog box, in the **Metric** and **SNMP OID** text boxes, type the metric and SNMP OID for the metric table (for example, **1.3.6.1.4.1.5951.4.1.1.41.1.5** and **11**).
7. Click **Add**.
8. Click **Close**. The metric table you created appears in the **Metric Tables** pane.

Unbinding Metrics from a Metrics Table

You can unbind metrics from a metrics table if the metrics need to be changed, or if you want to remove the metrics table entirely.

To unbind metrics from a metric table by using the NetScaler command line

At the NetScaler command prompt, type:

unbind metricTable <MetricTableName> <MetricType> <SNMPOID>

Example

unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5

To unbind metrics from a metric table by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Metric Tables**.
2. In the details pane, select the metric table from which you want to unbind the metrics (for example, **Table-Custom-1**), click **Open**.
3. In the **Configure Metric Table** dialog box, in the **Bound Metrics** list box, select the metric that you want to unbind from the table (for example, **1.3.6.1.4.1.5951.4.1.1.41.1.5**).
4. Click **Remove**, and then click **OK**.

You can view the detail of all configured metric tables, such as name and type, to determine whether the metric table is internal or created and configured.

Removing a Load Monitoring Metric Table

You can remove a metric table from the NetScaler configuration.

Note: Before you can remove a metric table, you must unbind all metrics from it.

To remove a metric table by using the NetScaler command line

At the NetScaler command prompt, type:

rm metricTable <MetricTableName>

Example

rm metricTable <Table-Custom-1>

To remove a metric table by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Metric Tables**.
2. In the details pane, select the metric table that you want to remove (for example, **Table-Custom-1**), and click **Remove**.
3. In the **Remove** dialog box, and then click **Yes**.

You can unbind a metric from a metric table to remove that metric from consideration.

Viewing Metrics Tables

You can view a metrics table and the metrics bound to it.

To view the metric tables by using the NetScaler command line

At the NetScaler command prompt, type:

show metricTable <MetricTableName>

Example

show metricTable Table-Custom-1

To view the metric tables by using the configuration utility

1. In the navigation pane, expand **Load Balancing**.
2. Click **Metric Tables**. The details of the available metric table appear on the **Metric Tables** pane.

Configuring Monitors in a Load Balancing Setup

To configure monitors on a Web site, you first decide whether to use a built-in monitor or create your own monitor. If you create a monitor, you can choose between creating a monitor based on a built-in monitor, or creating a custom monitor that uses a script that you write to monitor the service. (For more information about creating custom monitors, see Custom Monitors.) Once you have chosen or created a monitor, you then bind it to the appropriate service. The following conceptual diagram illustrates a basic load balancing setup with monitors.

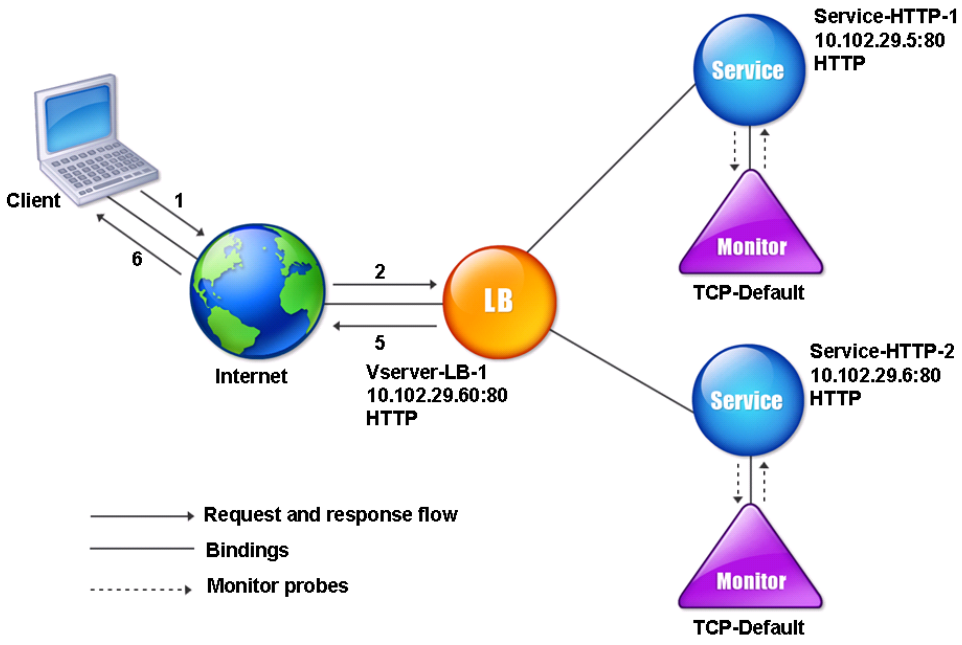


Figure 1. How Monitors Operate

As shown above, each service has a monitor bound to it. The monitor probes the load balanced server via its service. As long as the load balanced server responds to the probes, the monitor marks it UP. If the load balanced server should fail to respond to the designated number of probes within the designated time period, the monitor marks it DOWN.

Creating Monitors

The NetScaler appliance provides a set of built-in monitors. It also allows you to create custom monitors, either based on the built-in monitors or from scratch.

To create a monitor by using the NetScaler command line

At the NetScaler command prompt, type:

add lb mon <monitorName> <monitorType> [<interval>]

Example

add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2

Parameters for configuring monitors

monitorName

Name of the monitor. This alphanumeric string is required and cannot be changed after the monitor is created. The name must not exceed 31 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

monitorType

Type of monitor. The valid options for this parameter are: HTTP, PING, TCP, TCP-ECV, HTTP-ECV, UDP-ECV, DNS, FTP, RADIUS, USER, HTTP-INLINE, SIP-UDP, FTP-EXTENDED, SMTP, SNMP, NNTP, MYSQL, LDAP, POP3, LOAD, CITRIX-XML-SERVICE, CITRIX-WEB-INTERFACE, DNS-TCP, RTSP, ARP, CITRIX-AG, CITRIX-AAC-LOGINPAGE, CITRIX-AAC-LAS, and CITRIX-XD-DDC.

interval

Frequency at which the probe is sent to a service. The interval must be greater than the response time-out. Possible values: 1 millisecond-160 seconds. Default: 5 seconds.

To create a monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. On the **Monitors** pane, click **Add**.
3. In the **Create Monitor** dialog box, in the **Name** and **Interval** text boxes type the name and interval value of the monitor (for example, **monitor-HTTP-1** and **340**).
4. In the **Type** list, select the type of the monitor (for example, **HTTP**).
5. In the list next to the **Interval** text box, select **Seconds**.
6. Click **Create**, and then click **Close**. The monitor that you created appears in the **Monitors** pane.

Binding Monitors to Services

After creating a monitor, you bind it to a service. You can bind one or multiple monitors to a service. If you bind one monitor to a service, that monitor determines whether the service is marked UP or DOWN. If you bind multiple monitors to a service, the NetScaler appliance checks all monitors bound to that service using a calculation that you control, and marks the service UP or DOWN depending on the results.

Note: The destination IP address of a monitor probe can be different than the server IP address and port.

To bind a monitor to a service by using the NetScaler command line

At the NetScaler command prompt, type:

bind mon <MonitorName> <ServiceName>

Example

bind mon monitor-HTTP-1 Service-HTTP-1

To bind a monitor to a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service for which you want to bind the monitor (for example, **Service-HTTP-1**), and then click **Open**.
3. On the **Monitors** tab, in the **Available** list box, select the monitor you want to bind the service (for example, monitor-HTTP-1), and then click **Add**.
4. In the **Configured** box, click **OK**.

Modifying Monitors

You can modify the settings for any monitor that you created.

Note: Two sets of parameters apply to monitors: those that apply to all monitors, regardless of type, and those that are specific to a monitor type. For information on parameters for a specific monitor type, see the description for that type of monitor.

To modify an existing monitor by using the NetScaler command line

At the NetScaler command prompt, type:

set mon <MonitorName> <MonitorType> -**interval** <interval> -**resptimeout** <resptimeout>

Example

set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli

Parameters for modifying monitor settings

LRTM

State of the response time calculation of probes. Possible values: ENABLED and DISABLED. Default: DISABLED.

deviation

Deviation from the learned response time for dynamic response time monitoring. The maximum value is 348 minutes.

interval

Duration of the interval for which the NetScaler appliance waits before it marks the probe as failed. The response time-out must be less than the value specified in the interval parameter.

The UDP-ECV monitor type does not decide the probe failure using the response time-out. The appliance considers the probe as successful for the UDP-ECV monitor type when the server response matches the criteria that the send and receive options set, or

if the response is not received from the server.

The send option specifies the data that must be sent to the server in the probe, and the receive option specifies the server response criteria for the probe to succeed. The unreachable error from the service causes probe failure. The minimum value is 10 milliseconds. The maximum value is 159 seconds. The default value is 2 seconds.

resptimeout

Monitor response time-out threshold. If the response time for the monitoring probes exceeds the threshold, a trap is sent. The response time-out is given as a percentage. The minimum value is 1 and the maximum value is 100.

retries

Number of consecutive probe failures after which the NetScaler appliance determines the service as DOWN. Possible Values: 3 -127. Default: 3.

successRetries

Number of consecutive successful retries that are required to mark the state of the service as UP. For example, if you set the success retries to 3, when 3 probes succeed consecutively, the service is marked as UP. Possible Values: 1-32. Default: 1.

failureRetries

Number of failed probes that are required to mark the state of the service as DOWN. By default, the NetScaler appliance requires a specific number of consecutive retry failures to mark the state of the service as DOWN. The minimum value for this parameter is 0 and maximum value is 32. The default value is 0. For example, if you set the retries to 10 and the failure retries to 3, when 3 retry probes fail, the service is marked as DOWN.

alertRetries

The number of probe failures after which the NetScaler appliance generates an SNMP trap named MonProbeFailed. This parameter is closely linked to the Retries parameter. For example, if you set Retries to ten and SNMP Alert Retries to three, the appliance generates a MonProbeFailed trap when it detects a third probe failure. You can then take corrective action. However, if the problem is not corrected, the appliance marks the service as DOWN after the tenth probe failure.

For more information about SNMP traps, see the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667. You need to set the SNMP Alert Retries parameter to a value lower than the Retries parameter.

Note: The monitor probe failures need not be consecutive.

Possible Values: 0-32. Default: 0.

downTime

Wait duration until the next probe after the service is marked down. Possible Values: 10-160 seconds. Default: 30 seconds.

destIP

IP address to which the probe is sent. If the destination IP address is set to 0, the destination IP address is set to the bound service. Default: 0.0.0.0.

destPort

TCP/UDP port to which the probe is sent. If the destination port is set to 0, the destination port is the port of the service to which the monitor is bound. For a USER monitor, this port is the port sent in the HTTP request to the dispatcher. This option is ignored if the monitor is of the PING type. For information about user monitors, see Understanding User Monitors.

state

State of the monitor. If the monitor is disabled, this monitor type probe is not sent for the services. If the monitor is bound, the state of this monitor is not considered when the state of the service is determined. Possible values: ENABLED and DISABLED. Default: ENABLED.

reverse

Specifies whether the monitor is a reverse monitor. A reverse monitor marks a service as being down instead of up when probe criteria are satisfied and as being up instead of down when probe criteria are not satisfied. Possible values: YES and NO. Default: NO.

transparent

State of the monitor bound for transparent devices, such as firewalls, based on the responsiveness of the services behind them. If monitoring of transparent devices is enabled, the destination IP address must be specified. The probe is sent to the specified destination IP address using the MAC address of the transparent device. Possible values: YES and NO. Default: NO.

secure

State of the secure monitoring of services. SSL handshake is performed on the established TCP connection. Applicable for TCP-based monitors only. Possible values: YES and NO. Default: NO.

application

Name of the application that must be executed to check the state of the service.

sitePath

URL of the logon page.

To modify an existing monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, select the monitor that you want to modify (for example, **monitor-HTTP-1**), and then click **Open**.
3. On the **Standard Parameters** tab, in the **Interval** and **Response Time-out** text boxes, type the interval and response timeout values (for example, **50** and **20**).
4. In the list next to **Interval** text box, select the interval (for example, **Milli Seconds**).
5. In the list next to **Response Time-out** text box, select the interval (for example, **Milli Seconds**).
6. Click **OK**.

Enabling and Disabling Monitors

By default, monitors bound to services and service groups are enabled. When you enable a monitor, the monitor begins probing the services to which it is bound. If you disable a monitor bound to a service, the state the service is determined using the other monitors bound to the service. If the service is bound to only one monitor, and if you disable the monitor, the state of the service is determined using the default monitor.

To enable a monitor by using the NetScaler command line

At the NetScaler command prompt, type:

enable lb mon <monitorName>

Example

enable lb mon monitor-HTTP-1

To enable a monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. On the **Monitors** pane, select the monitor that you want to enable (for example, **monitor-HTTP-1**), and then click **Enable**.
3. In the **Enable** dialog box, click **Yes**.

To disable a monitor by using the NetScaler command line

At the NetScaler command prompt, type:

disable lb mon <monitorName>

Example

```
disable lb mon monitor-HTTP-1
```

To disable a monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. On the **Monitors** pane, select the monitor that you want to disable (for example, **monitor-HTTP-1**), and then click **Disable**.
3. In the **Disable** dialog box, click **Yes**.

Unbinding Monitors

You can unbind monitors from a service and service group. When you unbind a monitor from the service group, the monitors are unbound from the individual services that constitute the service group. When you unbind a monitor from a service or a service group, the monitor does not probe the service or the service group.

Note: When you unbind all user-configured monitors from a service or a service group, the default monitor is bound to the service and the service group. The default monitors then probes the service or the service groups.

To unbind a monitor from a service by using the NetScaler command line

At the NetScaler command prompt, type:

unbind mon <MonitorName> <ServiceName>

Example

unbind mon monitor-HTTP-1 Service-HTTP-1

To unbind a monitor from a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service from that you want to unbind the monitor (for example, **Service-HTTP-1**), click **Open**.
3. In the **Configure Service** dialog box, under **Configured**, select the monitor that you want to unbind from the service (for example, **monitor-HTTP-1**), and then click **Remove**.
4. Click **OK**.

Removing Monitors

After you unbind a monitor that you created from its service, you can remove that monitor from the NetScaler configuration. (If a monitor is bound to a service, it cannot be removed.)

Note: When you remove monitors bound to a service, the default monitor is bound to the service. You cannot remove default monitors.

To remove a monitor by using the NetScaler command line

At the NetScaler command prompt, type:

rm lb monitor <MonitorName> <MonitorType>

Example

rm lb monitor monitor-HTTP-1 HTTP

To remove a monitor by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. On the **Monitors** pane, select the monitor that you want to remove (for example, **monitor-HTTP-1**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Viewing Monitors

You can view the services and service groups that are bound to a monitor. You can verify the settings of a monitor to troubleshoot your NetScaler configuration. The following procedure describes the steps to view the bindings of a monitor to the services and service groups.

To view monitor bindings by using the NetScaler command line

At the NetScaler command prompt, type:

show lb monbindings <MonitorName>

Example

show lb monbindings monitor-HTTP-1

To view monitor bindings by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. On the **Monitors** pane, select the monitor for which you want to view the binding information (for example, **monitor-HTTP-1**), and then click **Show Bindings**. The binding information for the monitor that you selected appears in the **Binding Info for Monitor: monitor-HTTP-1** dialog box.

To view monitors by using the NetScaler command line

At the NetScaler command prompt, type:

show lb mon <MonitorName>

Example

```
show lb mon monitor-HTTP-1
```

To view monitors by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**. The details of the available monitors appear on the **Monitors** pane.

Closing Monitor Connections

The NetScaler appliance sends probes to the services through the monitors bound to the services. By default, the monitor on the NetScaler and the physical server follow the complete handshake procedure even for monitor probes. However, this procedure adds overhead to the monitoring process and may not be always necessary.

For the TCP monitors, you can configure the NetScaler to close a monitor-probe connection after receiving SYN-ACK from the service. To do so, set the value of the monitorConnectionClose parameter to RESET. If you want the monitor-probe connection to go through the complete procedure, set the value to FIN.

Note: The monitorConnectionClose setting is applicable to all the monitors bound to all the services configured on the NetScaler appliance.

To configure monitor-connection closure by using the NetScaler command line

At the NetScaler command prompt, type:

set lb parameter **-monitorConnectionClose** <monitor_conn_close_option>

Example

set lb parameter -monitorConnectionClose RESET

Parameter for configuring monitor-connection closure

monitorConnectionClose

Close (reset) a monitor-probe connection after receiving SYN-ACK from the service, or complete (finish) the handshake before closing the connection. Possible values: RESET, and FIN. Default: FIN.

To configure monitor-connection closure by using the NetScaler configuration utility

1. In the navigation pane, click **Load Balancing**.
2. Under **Settings**, click **Configure Load Balancing Parameters**.
3. In the **Configure Load Balancing Parameters** dialog box, for **Connection Close for Monitor**, select **FIN** or **RESET**.
4. Click **OK**.

Ignoring the Upper Limit on Client Connections for Monitor Probes

Depending on considerations such as the capacity of a physical server, you can specify a limit on the maximum number of client connections made to any service. If you have set such a limit on a service, the NetScaler appliance stops sending requests to the service when the threshold is reached and resumes sending connections to the service after the number of existing connections falls to within the limits. You can configure the NetScaler to skip this check when it sends monitor-probe connections to a service.

Note: You cannot skip the maximum-client-connections check for an individual service. If you specify this option, it applies to all the monitors bound to all the services configured on the NetScaler appliance.

To set the Skip MaxClients for Monitor Connections option by using the NetScaler command line

At the NetScaler command prompt, type:

set lb parameter **-monitorSkipMaxClient** (ENABLED|DISABLED)

Example

set lb parameter -monitorSkipMaxClient enabled

Parameter for skipping the maximum-client-connections check

monitorSkipMaxClient

For monitor-probe connections, ignore any maximum-client-connections limits that have been specified for the services being monitored. Possible values: ENABLED and DISABLED. Default: DISABLED.

To set the Skip MaxClients for Monitor Connections option by using the NetScaler configuration utility

1. In the navigation pane, click **Load Balancing**.
2. Under **Settings**, click **Configure Load Balancing Parameters**.
3. To ignore the upper limit on client connections for monitor probes, in the **Configure Load Balancing Parameters** dialog box, select the **Skip MaxClients for Monitor Connections** checkbox.
4. Click **OK**.

Configuring Support for Third-Party Load Balancers by Using SASP

A NetScaler appliance configured for Server/Application State Protocol (SASP) can work in conjunction with other load-balancing products. SASP accommodates load balancing based on the weights of the services. When SASP is implemented, a work load manager (WLM) agent runs on each server and relays performance data to the enterprise work load manager (EWLM). The EWLM uses this data to dynamically calculate the weight of each server, and then pushes the dynamically calculated weights to the appliance.

Note: SASP is not supported on NetScaler nCore builds.

The prerequisites for dynamic weight calculation are:

- The EWLM is configured as an entity within the appliance.
- A connection is established between the EWLM and the virtual server.
- The services bound to the virtual server are registered in the EWLM.

The following diagram shows how SASP facilitates load balancing decisions by using the WLM.

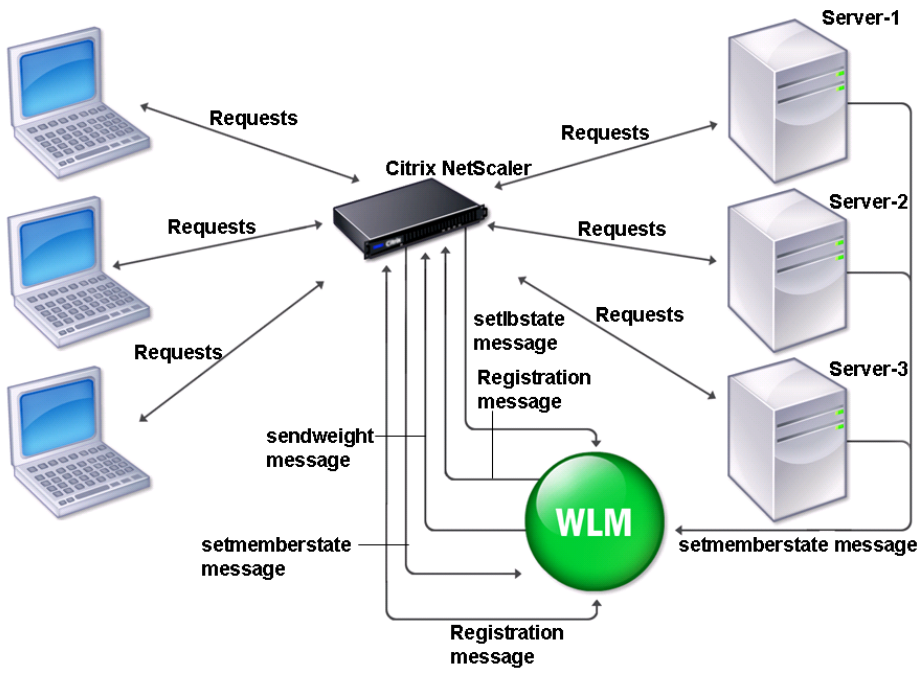


Figure 1. How SASP Load Balancing Works

The appliance and the EWLM communicate through a set of SASP messages. When the appliance is connected to the EWLM, a setlbstate message is sent to the EWLM to indicate that the connection is established. After the connection is established, the appliance sends a registration message to the EWLM to indicate that all services are registered with the EWLM. The EWLM responds with a registration success or failure message.

A connection is established only when a virtual server is bound to the WLM in the appliance. After all services are registered, the EWLM starts sending weight data to the appliance using the sendweight message. The WLM that is connected to each of the services sends the weight messages to the EWLM, as shown in the diagram above. The weight is calculated for the registered services only.

The appliance waits for the designated wait time (by default, two minutes) to receive the weight message from the EWLM. If the appliance receives the weight message within two minutes, the weight is dynamically calculated from the incoming weight message. If not, the appliance instead uses the user configured weights for making load balancing decisions.

If a service is disabled in the appliance, a setmemberstate message is sent to the EWLM conveying that the disabled service should not be considered for load balancing. The appliance sends a deregistration message to the EWLM to deregister or remove the disabled service. The EWLM responds with a deregistration success or failure message.

The following example shows the steps required to bind the services Service-HTTP-1 and Service-HTTP-2 to the virtual server Vserver-LB-1. Vserver-LB-1 forwards the client request to either of the two services Service-HTTP-1 or Service-HTTP-2. The appliance selects the service for each request using the least connection load balancing method. A workload manager Wlm-1 is created and bound to Vserver-LB-1.

The following diagram shows the load balancing entities and the values of the parameters.

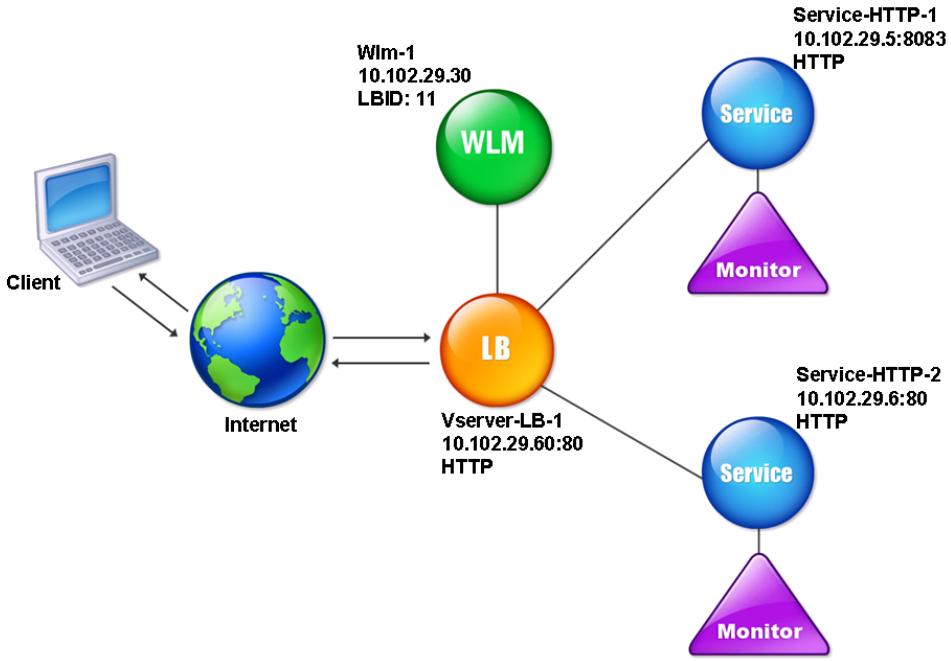


Figure 2. WLM Entity Model

Creating a Work Load Manager

You can create a work load manager to dynamically calculate the load on each service. The NetScaler uses the load data to select services for load balancing.

To create a work load manager by using the NetScaler command line

At the NetScaler command prompt, type:

add lb wlm <WLMName> <IPAddress> -**LBUID** <LBUniqueIdentifier>

Example

add lb wlm wlm-1 10.102.29.30 -LBUID 11

Parameters for configuring a work load manager

WLMName

Name of the work load manager. This alphanumeric string is required and cannot be changed after the work load manager is created. The name must not exceed 31 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

IPAddress

IP address of the work load manager.

LBUID

Unique identifier for the NetScaler to communicate to the work load manager.

port

Port of the work load manager. The port number must be a positive number and must not exceed 65535.

To create a work load manager by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Work Load Managers**.
2. On the **Work Load Managers** pane, click **Add**.
3. In the **Create Work Load Manager** dialog box, in the **Name**, **IP Address**, **LB Unique Identifier**, **Port**, and **Keep Alive Time-out** (minutes) text boxes, type the corresponding values (for example, **Wlm-1**, **10.102.29.30**, **11**, **80**, and **2**).
4. Click **Create**, and then click **Close**. The work load manager you created appears in the **Work Load Managers** pane.

Binding a Virtual Server to the Work Load Manager

The work load manager assigns a weight to the service on which it runs. The appliance requires a connection to balance the load on the services. When you bind a virtual server to a work load manager (WLM), the connection is established and the appliance uses the virtual server to balance the services.

To bind a virtual server to a work load manager by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb wlm <WLMName> <vServerName>

Example

bind lb wlm wlm-1 Vserver-LB-1

To bind a virtual server to a work load manager by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Work Load Managers**.
2. In the details pane, select the work load manager for which you want to bind the virtual server (for example, **Wlm-1**), and then click **Open**.
3. In the **Configure Work Load Manager** dialog box, under **Virtual Servers**, in the **Available** list box, select the virtual server that you want to bind to the work load manager (for example, **Vserver-LB-1**).
4. Click **Add**, and then click **OK**.

Managing the Work Load Manager

Managing the work load manager allows the NetScaler appliance to use the manually configured weights on the services. You can perform tasks such as removing or unbinding the work load manager from the virtual server.

Modifying the Work Load Manager

The NetScaler appliance periodically probes the work load manger. You can modify the time interval that the appliance uses to probe the WLM.

To modify a work load manager by using the NetScaler command line

At the NetScaler command prompt, type:

set lb wlm <WLMName> -**KATimeout** <TimeoutValue>

Example

set lb wlm wlm-1 -KATimeout 20

Parameter for modifying a work load manager

KATimeout

The idle time period after which the NetScaler probes the work load manager. The value ranges from 2 to 1440 minutes. The default value is 2 minutes and the maximum value is 1440 minutes.

To modify a work load manager by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Work Load Managers**.
2. In the details pane, select the workload manager that you want to modify (for example, **Wlm-1**), and then click **Open**.
3. In the **Configure Work Load Manager** dialog box, in the **Keep Alive Time-out (minutes)** text box, type the timeout value (for example, **20**).
4. Click **OK**.

Removing a Work Load Manager

You can remove a work load manager when you no longer need for the NetScaler appliance to dynamically calculate load on the service. When the work load manager is removed, the appliance uses the manually configured weights of the service to balance the load.

To remove a work load manager by using the NetScaler command line

At the NetScaler command prompt, type:

rm lb wlm <WLMName>

Example

rm lb wlm wlm-1

To remove a work load manager by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Work Load Managers**.
2. In the details pane, select the workload manager that you want to remove (for example, **Wlm-1**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Unbinding a Work Load Manager

When you unbind a work load manager from a virtual server, the virtual server reverts to using the manually configured weights on its services to select the service for each request.

To unbind a virtual server from a work load manager by using the NetScaler command line

At the NetScaler command prompt, type:

unbind lb wlm <WLMName> <vServerName>

Example

unbind lb wlm wlm-1 vserver-LB-1

To unbind a virtual server from a work load manager by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Work Load Managers**.
2. In the details pane, select the workload manager for which you want to unbind a virtual server (for example, **Wlm-1**), and then click **Open**.
3. In the **Configure Work Load Manager** dialog box, under **Virtual Servers**, in the **Configured** box, select the virtual server that you want to unbind from the work load manager (for example, **Vserver-LB-1**).
4. Click **Remove**, and then click **OK**.

Viewing a Work Load Manager

You can view the name, IP address, state, port, load balancing unique identifier, and keep-alive timeout for the configured work load managers. Viewing the details of the configuration is useful to check your setup.

To view work load managers by using the NetScaler command line

At the NetScaler command prompt, type:

show lb wlm <WLMName>

Example

show lb wlm wlm-1

To view work load managers by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Work Load Managers**.
2. In the details pane, view the details of the available work load managers.

Managing a Large Scale Deployment

The NetScaler appliance contains several features that are helpful when you are configuring a large load balancing deployment. Instead of configuring virtual servers and services individually, you can create groups of virtual servers and services. You can also create a range of virtual servers and services, and you can translate or mask virtual server and service IP addresses.

You can set persistence for a group of virtual servers. You can bind monitors to a group of services. Creating a range of virtual servers and services of identical type allows you to set up and configure those servers in a single procedure, which significantly shortens the time required to configure those virtual servers and services.

By translating or masking IP addresses, you can take down virtual servers and services, and make changes to your infrastructure, without extensive reconfiguration of your service and virtual server definitions.

Ranges of Virtual Servers and Services

When you configure load balancing, you can create ranges of virtual servers and services, eliminating the need to configure virtual servers and services individually. For example, you can use a single procedure to create three virtual servers with three corresponding IP addresses. When more than one argument uses a range, all of the ranges must be of the same size.

The following are the types of ranges you can specify when adding services and virtual servers to your configuration:

Numeric ranges. Instead of typing a single number, you can specify a range of consecutive numbers.

For example, you can create a range of virtual servers by specifying a starting IP address, such as 10.102.29.30, and then typing a value for the last byte that indicates the range, such as 34. In this example, five virtual servers will be created with IP addresses that range between 10.102.29.30 and 10.102.29.34.

Note: The IP addresses of the virtual servers and services must be consecutive.

Alphabetic ranges. Instead of typing a literal letter, you can substitute a range for any single letter, for example, [C-G]. This results in all letters in the range being included, in this case C, D, E, F, and G.

For example, if you have three virtual servers named **Vserver-x**, **Vserver-y**, and **Vserver-z**, instead of configuring them separately, you can type vserver [x-z] to configure them all.

Creating a Range of Virtual Servers

You create a range of virtual servers as described below.

To create range of virtual servers by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- add lb vserver <vServerName> <protocol> -**range** <rangeValue> <IPAddress> [<port>]
- add lb vserver <vServerName**[**<rangeValue>**]**> <protocol> <IPAddress**[**<rangeValue>**]**> [<port>]

Example

add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80

OR

> add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
 Done

Parameters for configuring virtual server ranges

vServerName

Name of the first virtual server in the range. If you use the second form of this command, you bracket the portion of the vServerName that contains the range.

Note: This command returns an error if the vServerName and IPAddress ranges that you define differ in number of entities.

protocol

The protocol of the virtual server.

rangeValue

The number of entities in the range that you are creating.

Note: Do not use -range and the [ ] range operator in the same command.

IPAddress

The IP address at the beginning of the range that you are defining. If you use the second form of this command, you bracket the portion of the IP address that contains the range.

port

The port of the virtual servers.

To create range of virtual servers by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add Range**.
3. In the **Create Virtual Server (Load Balancing) - Range** dialog box, in the **Name Prefix**, **IP Address Range**, and **Port** text boxes, type the virtual server name, IP address with which to begin the range, and port (for example, vserver, **10.102.29.30**, and **80**).
4. Select the **Network VServer** check box, and in **Range**, type the last value of the virtual server range (for example, **35**).
5. In the **Protocol** drop-down list box, select the protocol type (for example, **HTTP**).
6. Click **Create**, and then click **Close**. The range of virtual servers you created appears in the **Load Balancing Virtual Servers** pane.

Creating a Range of Services

You create a range of services as described below. If you specify a range for the service name, specify a range for the IP address too.

To create range of services by using the NetScaler command line

At the NetScaler command prompt, type the command:

add service <serviceName[<rangeValue>]> <IPAddress [<rangeValue>]> <protocol> <port>

Example

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
 Done
```

Parameters for configuring service ranges

serviceName

Name of the first service in the range. If you use the second form of this command, you bracket the portion of the serviceName that contains the range.

Note: This command returns an error if the serviceName and IPAddress ranges that you define differ in number of entities.

protocol

The protocol of the service.

rangeValue

The number of entities in the range that you are creating.

Note: Do not use -range and the [ ] range operator in the same command.

IPAddress

The IP address at the beginning of the range that you are defining. If you use the second form of this command, you bracket the portion of the IP address that contains the range.

port

The port of the services.

To create range of services by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add Range**.
3. In the **Create Service (Range)** dialog box, in the **IP Address Range** and **Port** text boxes, type the start value of the IP address range and the port (for example, **10.102.29.102**, and **80**).
4. In the text box next to the **IP Address Range** text box, type the last value of the last service (for example, **104**).
5. In the **Protocol** drop-down list box, select the protocol type (for example, **HTTP**).
6. Click **Create**, and then click **Close**. The range of services you created appears in the **Services** pane.

Configuring Service Groups

Configuring a service group enables you to manage a group of services as easily as a single service. For example, if you enable or disable any option, such as compression, health monitoring or graceful shutdown, for a service group, the option gets enabled for all the members of the service group.

After creating a service group, you can bind it to a virtual server, and you can add services to the group. You can also bind monitors to service groups.

The members of a service group can be identified by IP address or server name.

Using domain-name based service (DBS) group members is advantageous because you need not reconfigure the member on the NetScaler appliance if the IP address of the member changes. The appliance automatically senses such changes through the configured name server. This feature is particularly useful in cloud scenarios, where the service provider can change a physical server or change the IP address for a service. If you specify a DBS group member, the NetScaler learns the IP address dynamically.

You can bind both IP-based and DBS members to the same service group.

Note: If you use DBS service group members, make sure that either a name server is specified or a DNS server is configured on the NetScaler. A domain name will be resolved into an IP address only if the corresponding address record is present on the NetScaler or the name server.

Creating Service Groups

You can configure up to 4096 service groups on the NetScaler appliance.

To create a service group by using the NetScaler command line

At the NetScaler command prompt, type:

add servicegroup <ServiceGroupName> <Protocol>

Example

add servicegroup Service-Group-1 HTTP

Parameters for creating service groups

serviceGroupName

Name of the service group. This alphanumeric string is required and cannot be changed after the service group is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

Protocol

Type of service in this group. The valid options are: HTTP, TCP, FTP, UDP, SSL, SSL_TCP, SSL_BRIDGE, NNTP, DNS, RDP, and ANY

To create a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, click **Add**.
3. In the **Create Service Group** dialog box, in the **Service Group Name** text box, type name of the service group (for example, **Service-Group-1**).
4. In the **Protocol** list, select the protocol type (for example, **HTTP**).
5. Click **Create**, and then click **Close.** The service group you created appears in the **Service Groups** pane.

Binding a Service Group to a Virtual Server

When you bind a service group to a virtual server, the member services are bound to the virtual server.

To bind a service group to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

bind lb vserver <vServerName> <ServiceGroupName>

Example

bind lb vserver Vserver-LB-1 Service-Group-1

To bind a service group to a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind the service group (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Services Groups** tab.
4. In the **Active** column, select check box next to the service group that you want to bind to the virtual server (for example, **Service-Group-1**), and then click **OK**.

Binding a Member to a Service Group

Adding services to a service group enables the service group to manage the servers. You can add the servers to a service group by specifying the IP addresses or the names of the servers.

To add members to a service group by using the NetScaler command line

To configure a service group, at the NetScaler command prompt, type:

bind servicegroup <serviceGroupName> (<ipAddress> | <serverName>) <port>

Examples

bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a:888b 80

bind servicegroup CitrixEdu s1.citrite.net

To add members to a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group to which you want to bind members, and then click **Open**.
3. In the **Configure Service Group** dialog box, under **Specify Member(s)**, do one of the following:
   - To add a new IP based service group member, select **IP Based**.
   - To add a server-name based service group member, select **Server Based**.

   If you want to add a domain-name based service group member, select **Server Based**. With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.
4. If adding a new IP based member, in the **IP Address** text box, type the IP address. If the IP address uses IPv6 format, select the **IPv6** check box and then enter the address in the **IP Address** text box.

   Note: You can add a range of IP addresses. The IP addresses in the range must be consecutive. Specify the range by entering the starting IP address in the IP Address text box (for example, 10.102.29.30). Specify the end byte of the IP address range in the text box under Range (for example, 35). In the Port text box type the port (for example, 80), and then click **Add.**
5. Click **OK**.

Binding a Monitor to a Service Group

When you create a service group, the default monitor of the type appropriate for the group is automatically bound to it. Monitors periodically probe the servers in the service group to which they are bound and update the state of the service groups.

You can bind a different monitor of your own choice to the service group.

To bind monitor to a service group by using the NetScaler command line

At the NetScaler command prompt, type:

bind mon <MonitorName> <ServiceGroupName>

Example

bind mon monitor-HTTP-1 Service-Group-1

To bind monitor to a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group for which you want to bind monitors (for example, **Service-Group-1**), and then click **Open**.
3. On the **Monitors** tab, under **Available**, select a monitor name (for example, **ping**).
4. Click **Add**, and then click **OK**.

Managing Service Groups

You can change the settings of the services in a service group, and you can perform tasks such as enabling, disabling, and removing service groups. You can also unbind members from a service group.

Modifying a Service Group

You can modify attributes of service group members. You can set several attributes of the service group, such as maximum client, SureConnect, and compression. The attributes are set on the individual servers in the service group. You cannot set parameters on the service group such as transport information (IP address and port), weight, and server ID.

Note: A parameter you set for a service group is applied to the member servers in the group, not to individual services.

To modify a service group by using the NetScaler command line

At the NetScaler command prompt, type the following command with one or more of the optional parameters:

set servicegroup <ServiceGroupName> [-**type** <type>] [-**maxClient** <maxClient>] [-**maxReq** <maxReq>] [-**cacheable** (**YES**|**NO**)] [-**cip** (**ENABLED**|**DISABLED**)] [-**cipHeader** <cipHeader>] [-**usip** (**YES**|**NO**)] [-**sc** (**ON**|**OFF**)] [-**sp** (**ON**|**OFF**)] [-**cltTimeout** <cltTimeout>] [-**svrTimeout** <svrTimeout>] [-**cka** (**YES**|**NO**)] [-**TCPB** (**YES**|**NO**)] [-**CMP** (**YES**|**NO**)] [-**maxBandwidth** <maxBandwidth>] [-**maxThreshold** <maxThreshold>] [-**state** (**ENABLED**|**DISABLED**)] [-**downStateFlush** (**ENABLED**|**DISABLED**)

Example

set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES

Parameters for modifying service groups

type

Cache server supports the cache type option. Possible values: TRANSPARENT, REVERSE, and FORWARD.

maxClient

Maximum number of open connections to each service in the service group. The default value is 0. The maximum value is 4294967294.

maxReq

Maximum number of requests that can be sent over a persistent connection to a service in the service group. The default value is 0. The minimum value is 0. The maximum value is 65535.

cacheable

Whether a virtual server used for load balancing or content switching feature routes a request to the virtual server (used in transparent cache redirection) on the same appliance before sending it to the configured servers. The virtual server used for transparent cache redirection determines if the request is directed to the cache servers or the configured servers. Do not specify this argument if a cache type is specified. By default, this argument is disabled. Possible values: YES and NO. Default: NO.

cip

Enables or disables insertion of the Client IP header for services in the service group. Possible values: ENABLED and DISABLED. Default: DISABLED.

cipHeader

Client IP header. If client IP insertion is enabled and the client IP header is not specified, then the NetScaler sets the value of the Client IP header.

usip

Use of the client IP address as the source IP address while connecting to the server. By default, the appliance uses a mapped IP address for its server connection. However, with this option, you can tell the appliance to use the client's IP address when it communicates with the server. Possible values: yes and no. Default: no.

sc

The state of SureConnect on this service group. This parameter is supported for legacy purposes only; it has no effect, and the only valid value is OFF. Possible values: ON and OFF. Default: OFF.

sp

Whether surge protection needs to be enabled on this service group. Possible values: ON and OFF. Default: OFF

cltTimeout

Idle time in seconds after which the client connection is terminated. Default: 180. Maximum value: 31536000.

svrTimeout

Idle time in seconds after which the server connection is terminated. The default value is 360. The maximum value is 31536000.

CKA

State of the client keep-alive feature for the services in the service group. Possible values: YES and NO. Default: NO.

TCPB

State of the TCP Buffering feature for the services in the service group. Possible values: YES and NO. Default: NO.

CMP

State of the HTTP Compression feature for the services in the service group. Possible values: YES and NO. Default: NO.

maxBandwidth

Positive integer that identifies the maximum bandwidth in kilobits allowed for the services in the service group.

maxThreshold

Monitoring threshold. The default value is 0. The minimum value is 0 and maximum value is 65535.

state

State of the service group after it is added. Possible values: ENABLED and DISABLED. Default: ENABLED.

DownStateFlush

Delayed cleanup of connections on this service group. Possible values: ENABLED and DISABLED. Default: ENABLED.

Note: Any parameter you set on the service group is applied to the member servers in the group, not to individual services.

To modify a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group that you want to modify (for example, **Service-Group-1**), and then click **Open**.
3. Change any of the parameters described in "Parameters for modifying service groups," and then click **OK**.

Removing a Service Group

When you remove a service group, the servers bound to the group retain their individual settings and continue to exist on the NetScaler.

To remove a service group by using the NetScaler command line

At the NetScaler command prompt, type:

rm servicegroup <ServiceGroupName>

Example

rm servicegroup Service-Group-1

To remove a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group that you want to remove (for example, **Service-Group-1**), and then click **Remove**.
3. In the **Remove** dialog box, click **Yes**.

Unbinding a Member from a Service Group

When you unbind a member from the service group, the attributes set on the service group will no longer apply to the member that you unbound. The member services retains its individual settings, however, and continues to exist on the NetScaler.

To unbind members from a service group by using the NetScaler command line

At the NetScaler command prompt, type:

unbind servicegroup <ServiceGroupName> <IPAddress> [<port>]

Example

unbind servicegroup Service-Group-1 10.102.29.30 80

To unbind members from a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group from which you want to unbind members (for example, **Service-Group-1**), and then click **Open**.
3. In the **Configure Service Group** dialog box, in the **Configured Members** list box, select a service (for example, **10.102.29.30**).
4. Click **Remove**, and then click **OK**.

Unbinding a Service Group from a Virtual Server

When you unbind a service group from a virtual server, the member services are unbound from the virtual server and continue to exist on the NetScaler appliance.

To unbind a service group from a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

unbind lb vserver <vServerName> <ServiceGroupName>

Example

unbind lb vserver Vserver-LB-1 Service-Group-1

To unbind a service group from a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server from which you want to unbind the service group (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Services Group** tab.
4. Clear the **Active** check box next to the service group that you want to unbind from the virtual server (for example, **Service-Group-1**).
5. Click **OK**.

Unbinding Monitors from Service Groups

When you unbind a monitor from a service group, the monitor that you unbound no longer monitors the individual services that constitute the group.

To unbind a monitor from a service group using the NetScaler command line

At the NetScaler command prompt, type:

unbind mon <MonitorName> <ServiceGroupName>

Example

unbind mon monitor-HTTP-1 Service-Group-1

To unbind a monitor from a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group from which you want to unbind the monitor (for example, **Service-Group-1**), click **Open**.
3. In the **Configure Service Group** dialog box, click the **Monitors** tab.
4. Under **Configured**, select the monitor that you want to unbind from the service group (for example, **monitor-HTTP-1**), and then click **Remove**.
5. Click **OK**.

Enabling or Disabling a Service Group

When you enable a service group and the servers, the services belonging to the service group are enabled. Similarly, when a service belonging to a service group is enabled, the service group and the service are enabled. By default, service groups are enabled.

After disabling an enabled service, you can view the service using the configuration utility or the command line to see the amount of time that remains before the service goes DOWN.

To disable a service group by using the NetScaler command line

At the NetScaler command prompt, type:

disable servicegroup <ServiceGroupName>

Example

disable servicegroup Service-Group-1

To disable a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the **Service Groups** pane, select the service group that you want to disable (for example, **Service-Group-1**), and then click **Disable**.
3. In the **Wait Time** dialog box type the wait time value (for example, **30**).
4. Click **Enter**.

To enable a service group by using the NetScaler command line

At the NetScaler command prompt, type:

enable servicegroup <ServiceGroupName>

Example

enable servicegroup Service-Group-1

To enable a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the **Service Groups** pane, select the service group that you want to enable (for example, **Service-Group-1**), and then click **Enable**.
3. In the **Enable** dialog box, click **Yes**.

Viewing the Properties of a Service Group

You can view the following settings of the configured service groups: name, IP address, state, protocol, maximum client connections, maximum requests per connection, maximum bandwidth, and monitor threshold. Viewing the details of the configuration can be helpful for troubleshooting your configuration.

To view the properties of a service group by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to display the group properties or the properties and the group members:

- show servicegroup <ServiceGroupName>
- show servicegroup <ServiceGroupName> -**includemembers**

Example

show servicegroup Service-Group-1

To view the properties of a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, click the name of the service group whose properties you want to view, and then click **Open**.

Viewing Service Group Statistics

You can view service-group statistical data, such as rate of requests, responses, request bytes, and response bytes. The NetScaler appliance uses the statistics of a service group, such as these, to balance the load on the services.

To view the statistics of a service group by using the NetScaler command line

At the NetScaler command prompt, type:

stat servicegroup <ServiceGroupName>

Example

stat servicegroup Service-Group-1

To view the statistics of a service group by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. In the details pane, select the service group for which statistics you want to view (for example, **Service-Group-1**), and then click **Statistics**. The statistics of the service group you selected appear in a new window.

Load Balancing Virtual Servers Bound to a Service Group

In large-scale deployments, the same service group can be bound to multiple load balancing virtual servers. In such a case, instead of viewing each virtual server to see the service group it is bound to, you can view a list of all the load balancing virtual servers bound to a service group. You can view the following details of each virtual server:

- Name
- State
- IP address
- Port

To display the virtual servers bound to a service group by using the NetScaler command line

At the NetScaler command prompt, type the following command to display the virtual servers bound to a service group:

show servicegroupbindings <serviceGroupName>

Example

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
```

Parameter

serviceGroupName

Indicates the name of the service group whose bindings you want to view.

To display the virtual servers bound to a service group by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Service Groups**.
2. Select a service group, and then click **Show Bindings**. The **Binding details for Service Group** box then displays all the load balancing virtual servers bound to the selected service group.

Translating the IP Address of a Domain-Based Server

To simplify maintenance on the NetScaler appliance and on the domain-based servers that are connected to it, you can configure IP address masks and translation IP addresses. These functions work together to parse incoming DNS packets and substitute a new IP address for a DNS-resolved IP address.

When configured for a domain-based server, IP address translation enables the appliance to locate an alternate server IP address in the event that you take the server down for maintenance or if you make any other infrastructure changes that affect the server.

When configuring the mask, you must use standard IP mask values (a power of two, minus one) and zeros, for example, 255.255.0.0. Non-zero values are only permitted in the starting octets.

When you configure a translation IP for a server, you create a 1:1 correspondence between a server IP address and an alternate server that shares leading or trailing octets in its IP address. The mask blocks particular octets in the original server's IP address. The DNS-resolved IP address is transformed to a new IP address by applying the translation IP address and the translation mask.

For example, you can configure a translation IP address of 10.20.0.0 and a translation mask of 255.255.0.0. If a DNS-resolved IP address for a server is 40.50.27.3, this address is transformed to 10.20.27.3. In this case, the translation IP address supplies the first two octets of the new address, and the mask passes through the last two octets from the original IP address. The reference to the original IP address, as resolved by DNS, is lost. Monitors for all services to which the server is bound also report on the transformed IP address.

When configuring a translation IP address for a domain-based server, you specify a mask and an IP address to which the DNS-resolved IP address is to be translated.

Note: Translation of the IP address is only possible for domain-based servers. You cannot use this feature for IP-based servers. The address pattern can be based on IPv4 addresses only.

To configure a translation IP address for a server by using the NetScaler command line

At the NetScaler command prompt, type:

add server <serverName> <serverDomainName> -**translationIp** <translationIPAddress> -**translationMask** <netMask> -**state** <**ENABLED**|**DISABLED**>

Example

```
add server myMaskedServer www.example.com -translationIp 10.10.10.10 -translationMask 255.255.0.0 -stat
```

Parameters for configuring translation IP addresses

serverName

Name of the domain-based server.

serverDomainName

Server's domain name (for example, www.example.com).

Note that for IP address translation, the domain name is required.

translationIP

IP address (relevant octets only) to which the resolved ip for the server needs to be translated (for example, 11.12.0.0).

translationMask

Mask determines the number of bits in the translation IP address that are to be considered when applying the transformation.

For example, if you want an original server IP of 10.20.30.40 to be translated to 11.12.30.40, you could specify the mask 255.255.0.0.

To configure a translation IP address for a server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Servers**.
2. In the details pane, click **Add**.
3. In the **Create Server** dialog box, in the **Server Name** field, enter a name.
4. In the **IP Address / Domain Name** field, enter the server's domain name.

   Note: Do not enter an IP address if you are entering a mask.
5. In the **Translation IP Address** field, enter the IP address of a server on the same subnet.
6. In the **Translation Mask** field, enter a valid mask (for example, **255.255.0.0**).
7. Click **Create**.

Masking a Virtual Server IP Address

You can configure a mask and a pattern instead of a fixed IP address for a virtual server. This enables traffic that is directed to any of the IP addresses that match the mask and pattern to be rerouted to a particular virtual server. For example, you can configure a mask that allows the first three octets of an IP address to be variable, so that traffic to 111.11.11.198, 22.22.22.198, and 33.33.33.198 is all sent to the same virtual server.

By configuring a mask for a virtual server IP address, you can avoid reconfiguration of your virtual servers due to a change in routing or another infrastructure change. The mask allows the traffic to continue to flow without extensive reconfiguration of your virtual servers.

The mask for a virtual server IP address works somewhat differently from the IP pattern definition for a server described in Translating the IP Address of a Domain-Based Server. For a virtual server IP address mask, a non-zero mask is interpreted as an octet that is considered. For a service, the non-zero value is blocked.

Additionally, for a virtual server IP address mask, either leading or trailing values can be considered. If the virtual server IP address mask considers values from the left of the IP address, this is known as a forward mask. If the mask considers the values to the right side of the address, this is known as a reverse mask.

Note: The NetScaler appliance evaluates all forward mask virtual servers before evaluating reverse mask virtual servers.

When masking a virtual server IP address, you also need to create an IP address pattern for matching incoming traffic with the correct virtual server. When the appliance receives an incoming IP packet, it matches the destination IP address in the packet with the bits that are considered in the IP address pattern, and after it finds a match, it applies the IP address mask to construct the final destination IP address.

Consider the following example:

- Destination IP address in the incoming packet: 10.102.27.189
- IP address pattern: 10.102.0.0
- IP mask: 255.255.0.0
- Constructed (final) destination IP address: 10.102.27.189.

In this case, the first 16 bits in the original destination IP address match the IP address pattern for this virtual server, so this incoming packet is routed to this virtual server.

If a destination IP address matches the IP patterns for more than one virtual server, the longest match takes precedence. Consider the following example:

- Virtual Server 1: IP pattern 10.10.0.0, IP mask 255.255.0.0

- Virtual Server 2: IP pattern 10.10.10.0, IP mask 255.255.255.0
- Destination IP address in the packet: 10.10.10.45.
- Selected virtual server: Virtual Server 2.

The pattern associated with Virtual Server 2 matches more bits than that associated with Virtual Server 1, so IPs that match it will be sent to Virtual Server 2.

Note: Ports are also considered if a tie-breaker is required.

To configure a virtual server IP address mask by using the NetScaler command line

At the NetScaler command prompt, type:

add lb vserver <vServerName> http -**ipPattern** <ipAddressPattern> -**ipMask** <ipMask> <listenPort>

Example

Pattern matching based on prefix octets:

add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask 255.255.0.0 80

Pattern matching based on trailing octets:

add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask 0.0.255.255 80

Modify a pattern-based virtual server:

set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255

Parameters for masking virtual server IP addresses

name

Name of the load balancing virtual server.

http

Value of HTTP

port

Listen port for the virtual server.

Pattern Based

(Configuration utility only.) Option to select if the virtual server is to be pattern-based.

ippattern

IP address pattern for the virtual server. You must supply either the initial or the trailing octets (for example, 11.11.00.00).

ipmask

Network mask for the IP address. Non-zero values indicate the IP address octets that are to be passed through. (For example, for an IP address pattern of 11.11.00.00, you might specify a mask of 255.255.0.0).

Note: You cannot convert a virtual server with a fixed IP address to a pattern-based virtual server, and you cannot convert a pattern-based virtual server to one with a fixed IP address.

To configure a virtual server IP address mask by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server** dialog box, in the **Name** field, enter a name.
4. In the **Protocol** field, select the protocol (for example, **HTTP**).
5. In the **Port** field, enter the listen port.
6. In the **IP Pattern** field, enter a pattern for an IP address (for example, **11.11.0.0**). The fixed part of the pattern must be entered in contiguous octets. Enter zeros for the pattern values that can vary in the IP address.
7. In the **IP Mask** field, enter a standard network mask (for example, **255.255.0.0**). Use non-zero mask values for the portion of the IP address that constitutes the fixed part of the pattern.

Configuring Load Balancing for Commonly Used Protocols

In addition to Web sites and Web-based applications, other types of network-deployed applications that use other common protocols often receive large amounts of traffic and therefore benefit from load balancing. Several of these protocols require specific configurations for load balancing to work properly. Among them are FTP, DNS, SIP, and RTSP.

If you configure your NetScaler appliance to use domain names for your servers rather than IPs, you may also need to set up IP translation and masking for those servers.

Load Balancing for a Group of FTP Servers

The NetScaler appliance can be used to load balance FTP servers. FTP requires that the user initiate two connections on two different ports to the same server: the control connection, through which the client sends commands to the server, and the data connection, through which the server sends data to the client. When the client initiates an FTP session by opening a control connection to the FTP server, the appliance uses the configured load balancing method to select an FTP service, and forwards the control connection to it. The load balanced FTP server then opens a data connection to the client for information exchange.

The following diagram describes the topology of a load balancing configuration for a group of FTP servers.

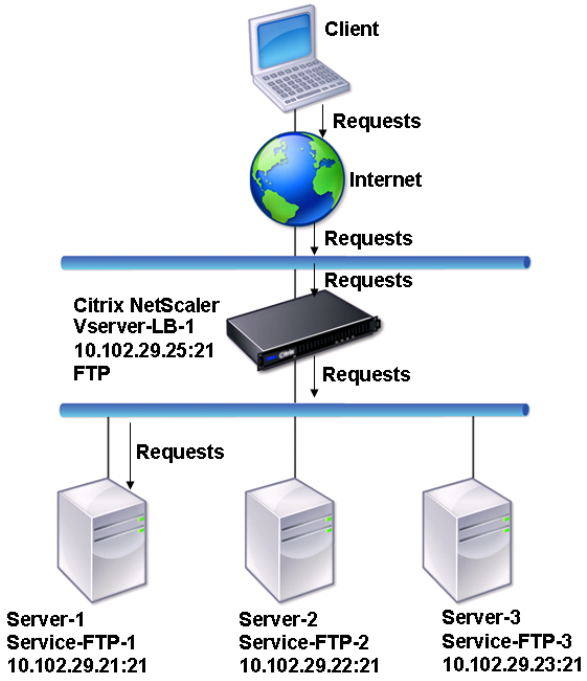


Figure 1. Basic Load Balancing Topology for FTP Servers

In the diagram, the services Service-FTP-1, Service-FTP-2, and Service-FTP-3 are bound to the virtual server Vserver-LB-1. Vserver-LB-1 forwards the client's connection request to one of the services using the least connection load balancing method. Subsequent requests are forwarded to the service that the appliance initially selected for load balancing.

The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
Services	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP
Monitors	FTP	None	None	None

The following diagram shows the load balancing entities, and the values of the parameters that need to be configured on the appliance.

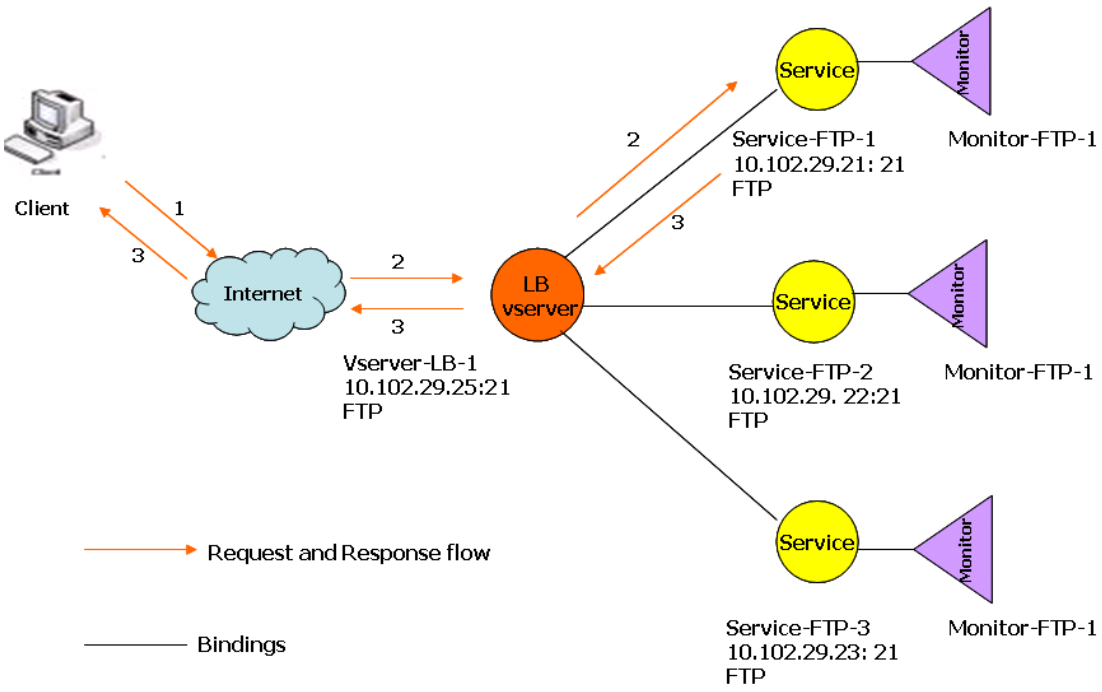


Figure 2. Load Balancing FTP Servers Entity Model

The appliance can also provide a passive FTP option to access FTP servers from outside of a firewall. When a client uses the passive FTP option and initiates a control connection to the FTP server, the FTP server also initiates a control connection to the client. It then initiates a data connection to transfer a file through the firewall.

To create services and virtual servers of type FTP, see Setting Up Basic Load Balancing. Name the entities and set the parameters to the values described in the columns of the previous table. When you configure a basic load balancing setup, a default monitor is bound to the services.

Next, bind the FTP monitor to the services by following the procedure described in the section Binding Monitors to Services.

To create FTP monitors by using the NetScaler command line

At the NetScaler command prompt, type:

add lb monitor <FTPMonitorName> -**interval** <Interval> -**userName** <UserName> -**password** <Password>

Example

add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password User

To create FTP monitors by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.
3. On the **Standards Parameters** tab, in the **Name** and **Interval** text boxes, type the monitor name and interval (for example, **monitor-FTP-1** and **340**).
4. In the **Type** list, select **FTP**.
5. On the **Special Parameters** tab, in the **User Name** and **Password** text boxes, type **User**.
6. Click **Create**, and then click **Close**. The monitor that you created appears in the **Monitors** pane.

Load Balancing DNS Servers

When you request DNS resolution of a domain name, the NetScaler appliance uses the configured load balancing method to select a DNS service. The DNS server to which the service is bound then resolves the domain name and returns the IP address as the response. The appliance can also cache DNS responses and use the cached information to respond to future requests for resolution of the same domain name. Load balancing DNS servers improves DNS response times.

For more information about DNS and caching DNS records, see Domain Name System.

The following diagram describes the topology of a load balancing configuration that load balances a group of DNS services.

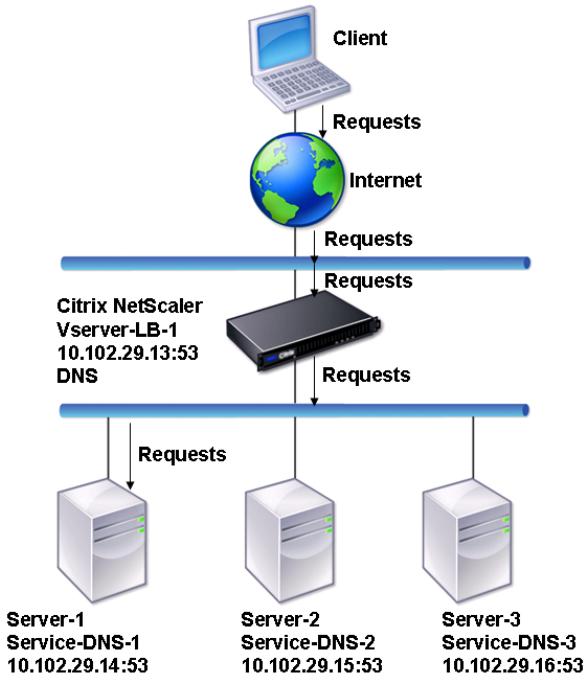


Figure 1. Basic Load Balancing Topology for DNS Servers

In the diagram, the services Service-DNS-1, Service-DNS-2, and Service-DNS-3 are bound to the virtual server Vserver-LB-1. The virtual server Vserver-LB-1 forwards client requests to a service using the least connection load balancing method. The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.13	53	DNS

Services	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
Monitors	monitor-DNS-1	None	None	None

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

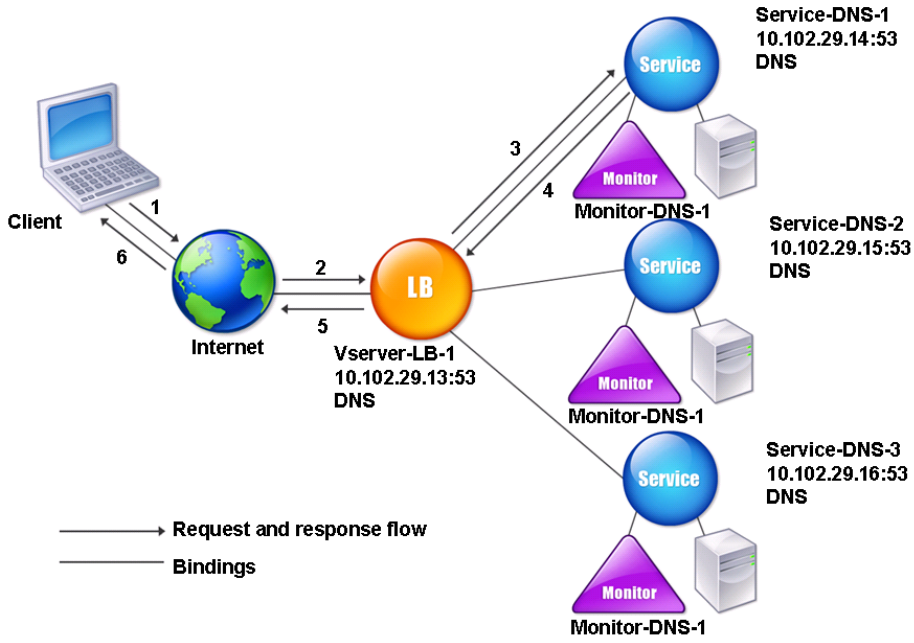


Figure 2. Load Balancing DNS Servers Entity Model

To configure a basic DNS load balancing setup, see Setting Up Basic Load Balancing. Follow the procedures to create services and virtual servers of type DNS, naming the entities and setting the parameters using the values described in the previous table. When you configure a basic load balancing setup, the default ping monitor is bound to the services. For instructions on binding a DNS monitor to DNS services, you can also see Binding Monitors to Services.

The following procedure describes the steps to create a monitor that maps a domain name to the IP address based on a query.

To configure DNS monitors by using the NetScaler command line

At the NetScaler command prompt, type:

add lb monitor <monitorName> DNS -**query** <domainName> -**queryType** <**Address**|**ZONE**> -**IPAddress** <ipAddress>

Example

add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType Address -IPAddress 10.102.29.66

add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType Address -IPAddress 1000:0000:0000:0

To configure DNS monitors by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor** dialog box, in the **Name** and **Interval** text boxes, type a monitor name and a monitoring interval (for example, **monitor-DNS-1** and **340**, respectively).
4. Select the unit of time for the interval in the drop-down menu.
5. In the **Type** list, select **DNS**.
6. Click the **Special Parameters** tab, in the **Query** text box type the domain name query to send to the DNS service (for example, **www.mycompany.com**), and in the **Query Type** list box, select **ADDRESS** or **ZONE**.
7. In the text box below the **Query Type** list box, type an IP address that is to be checked against the response to the DNS monitoring query (for example, **10.102.29.66**), and click **Add**.

   Note: If you want to enter an IPv6 address, select the IPv6 check box before entering the address.
8. Click **Create**, and then click **Close**. The monitor that you created appears in the **Monitors** pane.

Load Balancing Domain-Name Based Services

When you create a service for load balancing, you can provide an IP address. Alternatively, you can create a server using a domain name. The server name (domain name) can be resolved using an IPv4 or IPv6 name server, or by adding an authoritative DNS record (A record for IPv4 or AAAA record for IPv6) to the NetScaler configuration.

When you configure services with domain names instead of IPs, if you change the IP address of a server in your load balancing setup, the name server resolves the domain name to the new IP address. The monitor runs a health check on the new IP address, and updates the service IP address only when the IP address is found to be healthy.

Note: When you change the IP address of a server, the corresponding service is marked down for the first client request. The name server resolves the service IP address to the changed IP address for the next request, and the service is marked UP.

Domain-name based services have the following restrictions:

- The maximum domain name length is 255 characters.
- The Maximum Client parameter is used to configure a service that represents the domain name-based server. For example, a maxClient of 1000 is set for the services bound to a virtual server. When the connection count at the virtual server reaches 2000, the DNS resolver changes the IP address of the services. However, because the connection counter on the service is not reset, the virtual server cannot take any new connections until all the old connections are closed.
- When the IP address of the service changes, persistence is difficult to maintain.
- If the domain name resolution fails due to a timeout, the appliance uses the old information (IP address).
- When monitoring detects that a service is down, the appliance performs a DNS resolution on the service (representing the domain name-based server) to obtain a new IP address.
- Statistics are collected on a service and are not reset when the IP address changes.
- If a DNS resolution returns a code of “name error” (3), the appliance marks the service down and changes the IP address to zero.

When the appliance receives a request for a service, it selects the target service. This way, the appliance balances load on your services. The following diagram describes the topology of a load balancing configuration that load balances a group of domain-name based servers (DBS).

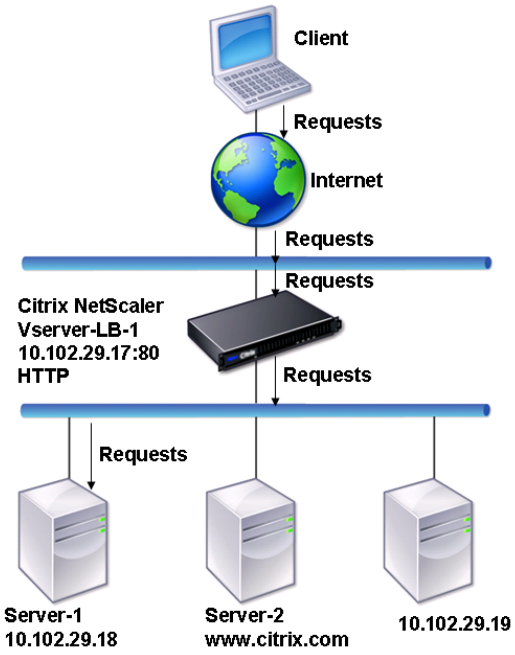


Figure 1. Basic Load Balancing Topology for DBS Servers

The services Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 are bound to the virtual server Vserver-LB-1. The vserver Vserver-LB-1 uses the least connection load balancing method to choose the service. The IP address of the service is resolved using the name server Vserver-LB-2.

The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.17	80	HTTP
	Vserver-LB-2	10.102.29.20	53	DNS
Servers	server-1	10.102.29.18	80	HTTP
	server-2	www.citrix.com	80	HTTP
Services	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
Monitors	Default	None	None	None
Name Server	None	10.102.29.19	None	None

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

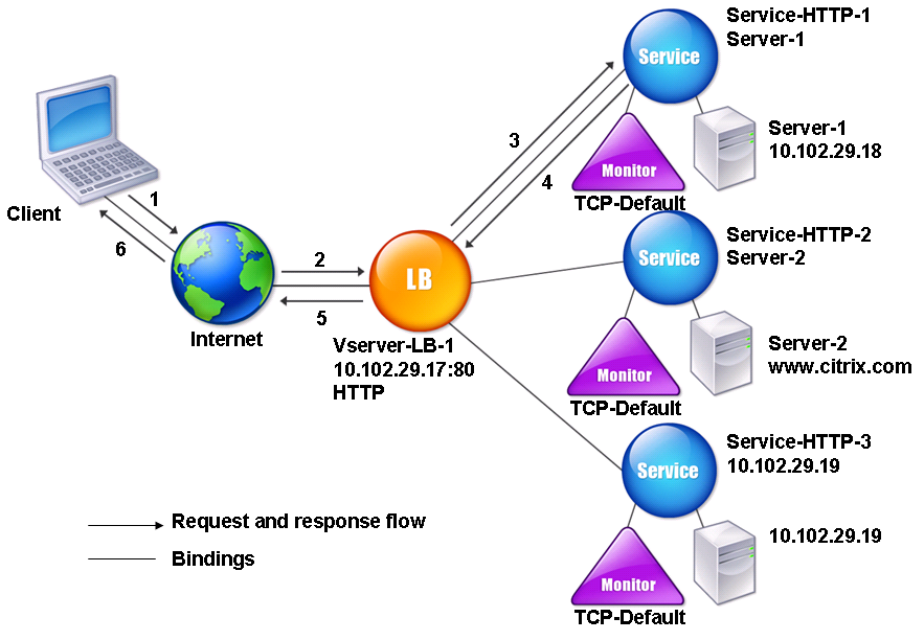


Figure 2. Load Balancing DBS Servers Entity Model

To configure a basic load balancing setup, see Setting Up Basic Load Balancing. Create the services and virtual servers of type HTTP, and name the entities and set the parameters using the values described in the previous table.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

To add a name server by using the NetScaler command line

At the NetScaler command prompt, type:

add dns nameServer <vServerName>

Example

add dns nameServer Vserver-LB-2

To add a name server by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Name Servers**.
2. In the details pane, click **Add**.
3. In the **Create Name Server** dialog box, select **DNS Virtual Server**.
4. In the **DNS Virtual Server** drop-down list, select the server name (for example, **Vserver-LB-2**).

   Note: Click **New** if you want to create a new load balancing vserver. The **Create Virtual Server (Load Balancing)** dialog box appears.
5. Click **Create**, and then click **Close**.

You can also add an authoritative name server that resolves the domain name to an IP address. For more information about configuring name servers, see Domain Name System.

Load Balancing a Group of SIP Servers

The NetScaler appliance can load balance SIP servers to improve the performance of a VoIP system. You can also enable RPORT on the appliance. For more information about RPORT, see the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671. The following diagram describes the topology of a load balancing setup configured to load balance a group of SIP servers.

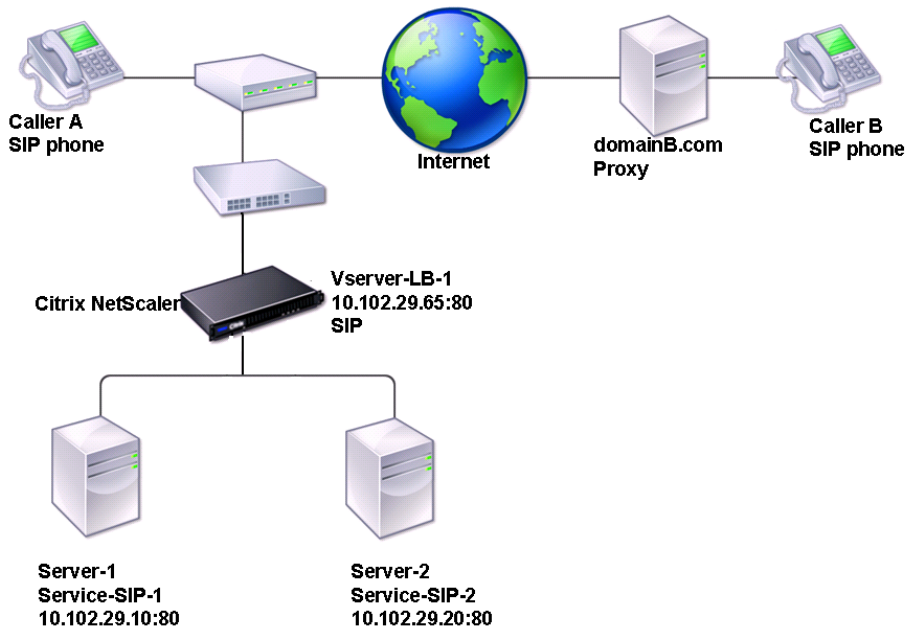


Figure 1. SIP Load Balancing Topology

In the example, the services Service-SIP-1 and Service-SIP-2 are bound to the virtual server Vserver-LB-1. The following table lists the names and values of the entities that you need to configure on the appliance in inline mode (also called two-arm mode).

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.65	80	SIP-UDP
Services	Service-SIP-1	10.102.29.10	80	SIP-UDP
	Service-SIP-2	10.102.29.20	80	SIP-UDP
Monitors	Default	None	80	SIP-UDP

The following diagram shows the load balancing entities and the values of the parameters to be configured on the appliance.

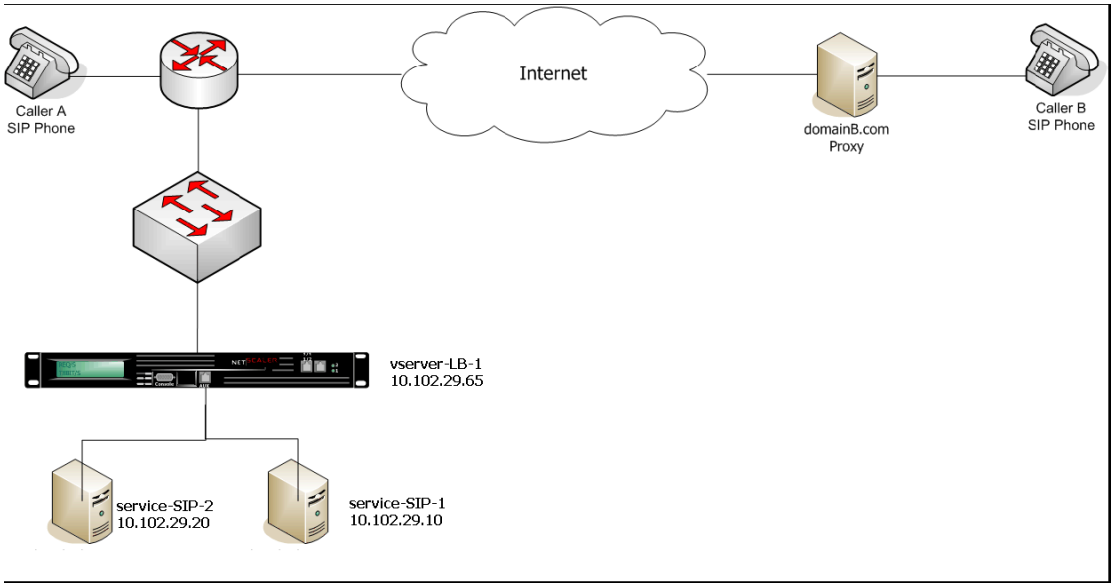


Figure 2. Load Balancing SIP Servers Entity Model

To configure a basic load balancing setup for SIP, see Setting Up Basic Load Balancing. You create services and virtual servers of type SIP-UDP, naming the entities and setting the parameters as described in the previous table. You must then configure RNAT.

To configure RNAT by using the NetScaler command line

At the NetScaler command prompt, type:

add route <network> <netmask> <gateway>

Example

add route 10.102.29.0 255.255.255.0 10.102.29.50

Parameters for configuring RNAT

network

The first IP in the network to which you are adding a route.

netmask

The network mask of the network to which you are adding a route.

gateway

The gateway of the network to which you are adding a route.

To configure RNAT by using the configuration utility

1. In the navigation pane expand **Network**, expand **Routing**, and then click **Routes**.
2. In the details pane, click **Add**.
3. In the **Create Route** dialog box, in the **Network**, **Netmask**, and **Gateway IP** text boxes type the network, netmask, and gateway of the network to which you are adding a route (for example, **10.102.29.0**, **255.255.255.0**, and **10.102.29.50**).
4. Click **Create**, and then click **Close**.

After you configure RNAT, the appliance sends SIP responses to the IP address and port that the client uses to send the request. The appliance also adds the RPORT tag in the VIA header of the message. The appliance compares the values of the source and destination ports of the request packets with the RNAT source port and RNAT destination port. If one of the values matches, the appliance updates the VIA header with the RPORT setting.

You must enable this setting when RPORT is not configured on either client.

To configure SIP parameters by using the NetScaler command line

At the NetScaler command prompt, type:

set sipParameters **-rnatSrcPort** <rnatSrcPort> **- sip503RateThreshold** <sip503_rate_threshold_value>

Example

set sipParameters -rnatSrcPort 5060 -sip503RateThreshold 1000

Parameters for configuring the SIP parameters

rnatSrcPort

The RNAT source port. Minimum value: 1, Maximum value: 65534.

sip503RateThreshold

Indicates the maximum number of SIP 503 error messages the virtual server will send in ten milliseconds. Minimum value: 0, Maximum value: 65535. Default: 100.

To configure SIP parameters by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. On the **Load Balancing** landing page, under **Settings**, click **Change SIP settings**.
3. In the **Set SIP Parameters** dialog box, in the **RNAT source port** box, type a value.
4. In the **Set SIP Parameters** dialog box, in the **SIP503 Rate Threshold** box, type a value.
5. Select the **Enable Add RPort VIP** check box, and then click **OK**.

Load Balancing RTSP Servers

The NetScaler appliance can balance load on RTSP servers to improve the performance of audio and video streams over networks. The following diagram describes the topology of an load balancing setup configured to load balance a group of RTSP servers.

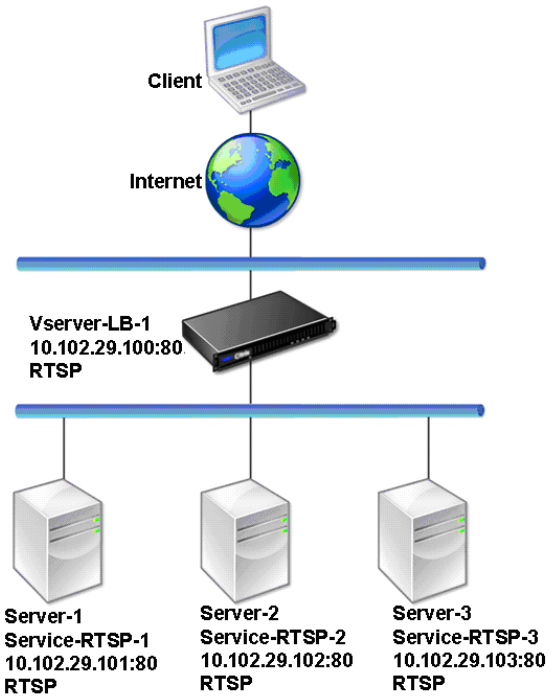


Figure 1. Load Balancing Topology for RTSP

In the example, the services Service-RTSP-1, Service-RTSP-2, and Service-RTSP-3 are bound to the virtual server Vserver-LB-1. The following table lists the names and values of the example entities.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.100	554	RTSP
Services	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
Monitors	Monitor-RTSP-1	None	554	RTSP

The following diagram shows the load balancing entities used in RTSP configuration.

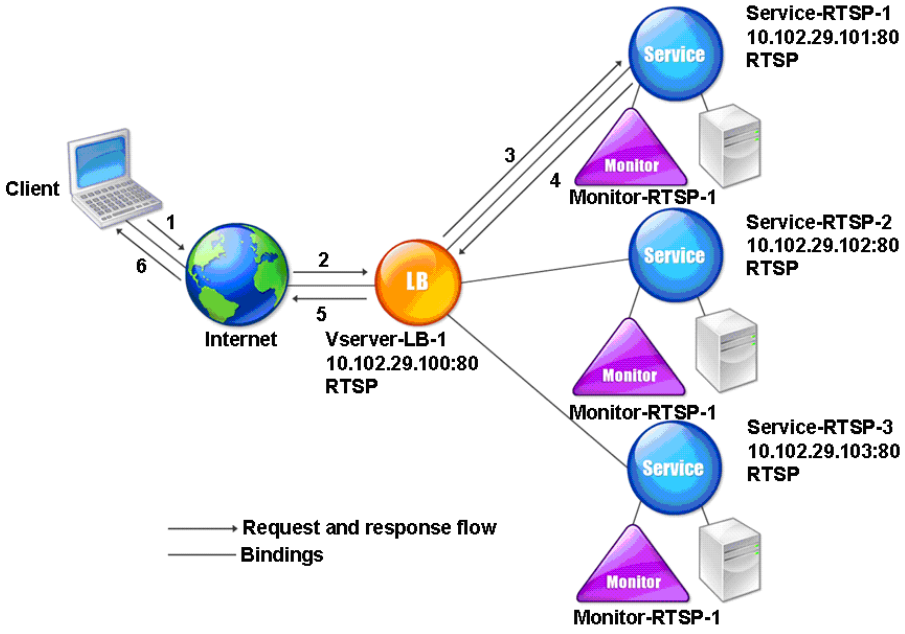


Figure 2. Load Balancing RTSP Servers Entity Model

To configure a basic load balancing setup for RTSP serveers, see Setting Up Basic Load Balancing. Create services and virtual servers of type RTSP. When you configure a basic load balancing setup, the default TCP-default monitor is bound to the services. To bind an RTSP monitor to these services, see Binding Monitors to Services. The following procedure describes how create a monitor that checks RTSP servers.

To configure RTSP monitors by using the NetScaler command line

At the NetScaler command prompt, type:

add lb monitor <NameOfMonitor> <TypeOfMonitor>

Example

add lb monitor Monitor-RTSP-1 RTSP

To configure RTSP monitors by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor** dialog box, in the **Name** and **Interval** text boxes, type the name and probing interval of a monitor (for example, **Monitor-RTSP-1** and **340**).
4. In the **Type** list, select the type of the monitor (for example, **RTSP**).
5. Click **Create**, and then click **Close**.

Load Balancing of Remote Desktop Protocol (RDP) Servers

Note: This feature is supported only on NetScaler nCore builds.

Remote Desktop Protocol (RDP) is a multichannel-capable protocol that allows for separate virtual channels for carrying presentation data, serial device communication, licensing information, highly encrypted data (keyboard and mouse activity), and so on.

RDP is used for providing a graphical user interface to another computer on the network. RDP is used with Windows terminal servers for providing fast access with almost real-time transmission of mouse movements and key presses even over low-bandwidth connections.

When multiple terminal servers are deployed to provide remote desktop services, the NetScaler appliance provides load balancing of the terminal servers (Windows 2003 and 2008 Server Enterprise Editions). In some cases, a user who is accessing an application remotely may want to leave the application running on the remote machine but shut down the local machine. The user therefore closes the local application without logging out of the remote application. After reconnecting to the remote machine, the user should be able to continue with the remote application. To provide this functionality, the NetScaler RDP implementation honors the routing token (cookie) set by the Terminal Services Session Directory or Broker so that the client can reconnect to the same terminal server to which it was connected previously. The Session Directory, implemented on Windows 2003 Terminal Server, is referred to as Broker on Windows 2008 Terminal Server.

When a TCP connection is established between the client and the load balancing virtual server, the NetScaler applies the specified load balancing method and forwards the request to one of the terminal servers. The terminal server checks the session directory to determine whether the client has a session running on any other terminal server in the domain.

If there is no active session on any other terminal server, the terminal server responds by serving the client request, and the NetScaler forwards the response to the client.

If there is an active session on any other terminal server, the terminal server that receives the request inserts a cookie (referred to as routing token) with the details of the active session and returns the packets to the NetScaler, which returns the packet to the client. The server closes the connection with the client. When the client retries to connect, the NetScaler reads the cookie information and forwards the packet to the terminal server on which the client has an active session.

The user on the client machine experiences a continuation of the service and does not have to take any specific action.

Note: The Windows Session Directory feature requires the Remote Desktop client that was first released with Windows XP. If a session with a Windows 2000 or Windows NT 4.0 Terminal Server client is disconnected and the client reconnects, the server with which the connection is established is selected by the load balancing algorithm.

The following diagram describes RDP load balancing.

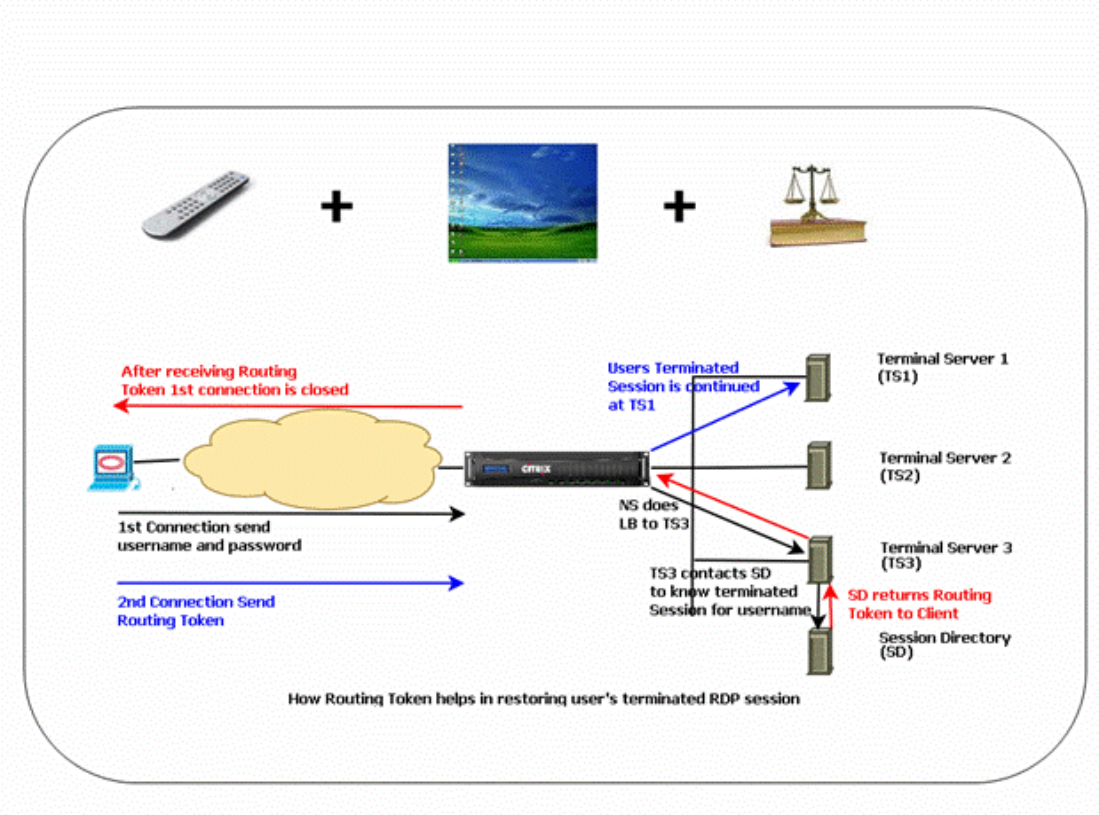


Figure 1. Load Balancing Topology for RDP

Note: When an RDP service is configured, session cookie persistence is automatically maintained. You need not enable persistence explicitly.

Ensure that the disconnected RDP sessions are cleared on the terminal servers at the backend to avoid flapping between two terminal servers when an RDP session is disconnected without logging out. For more information, see http://technet.microsoft.com/en-us/library/cc758177(WS.10).aspx#BKMK_2

When you add an RDP service, by default, NetScaler adds a monitor of the type TCP and binds it to the service. The default monitor is a simple TCP monitor that checks whether or not a listening process exists at the 3389 port on the server specified for the RDP service. If there is a listening process at 3389, NetScaler marks this service as UP and if there is no listening process, it marks the service as DOWN.

For more efficient monitoring of an RDP service, in addition to the default monitor, you can configure a script monitor that is meant for the RDP protocol. When you configure the scripting monitor, the NetScaler opens a TCP connection to the specified server and sends an RDP packet. The monitor marks the service as UP only if it receives a confirmation of the connection from the physical server. Therefore, from the scripting monitor, the NetScaler can know whether the RDP service is ready to service a request.

The monitor is a user-type monitor and the script is located on the NetScaler at /nsconfig/monitors/nsrdp.pl. When you configure the user monitor, the NetScaler runs the script automatically. To configure the scripting monitor, add the monitor and bind it to the RDP service.

To configure RDP load balancing, create services of type RDP and bind them to an RDP virtual server.

To configure RDP load balancing services by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an RDP load balancing setup and verify the configuration:

add service <serviceName> <serverName> <serviceType> <port>

Note: Repeat the above command to add more services.

Example

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
       State: UP
…
          Server Name: 10.102.27.182
          Server ID : 0          Monitor Threshold : 0
       Down state flush: ENABLED
…
1)     Monitor Name: tcp-default
            State: UP       Weight: 1
…
            Response Time: 4.152 millisec
 Done
```

Parameters for configuring a service

serviceName

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or a letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Name or IP address of the server in IPv4 format.

serviceType

The service type must be RDP.

port

The default port, 3389, must be used.

To configure RDP load balancing services by using the NetScaler configuration utility

1. In the **navigation** pane, expand **Load Balancing** and then click **Services**.
2. In the **details** pane, click **Add**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a service" as shown:
   - **Service Name*** —serviceName
   - **Protocol*** —serviceType
   - **Server*** —serverName
   - **Port*** —port

   *A required parameter
4. Click **Create**.
5. Create all the RDP services to be load balanced.
6. From the **services** pane, open the service you added, and verify the addition.

To configure an RDP load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an RDP load balancing virtual server and verify the configuration:

- add lb vserver <vServerName> <serviceType> <ipAddress> <port>
- bind lb vserver <vServerName> <serviceName>

  Bind all the RDP services to be load balanced to the virtual server.

Example

This example has two RDP services bound to the RDP virtual server.

```
> add lb vs v1 rdP 10.102.27.186 3389
Done

> bind lb vs v1 ser1
service "ser1" bound
> bind lb vs v1 ser2
service "ser2" bound
Done

>sh lb vs v1
v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
State: UP
…
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION
  Current Method: Round Robin, Reason: A new service is bound
Mode: IP
Persistence: NONE
  L2Conn: OFF

1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
```

Parameters for configuring a virtual server

vServerName

Name of the virtual server that is associated with the service. The name must not exceed 127 characters, and the leading character must be a number or a letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ipAddress

IP address of the virtual server in the IPv4 format.

serviceType

The service type must be RDP.

port

The default port, 3389, must be used.

To configure an RDP load balancing virtual server by using the NetScaler configuration utility

1. In the **navigation pane**, expand **Load Balancing** and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a virtual server" as shown:
   - **Name***—vServerName
   - **IP Address***—ipAddress
   - **Protocol***—serviceType
   - **Port***—port

   *A required parameter
4. In the **Services** tab, select the services to be bound to the virtual server by checking the service names.
5. Click **Create**.
6. In the **Load Balancing Virtual Servers** pane, select the RDP virtual server you configured, and then click **Open** to verify the configuration.

To configure a scripting monitor for RDP services by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- add lb monitor <rdpMonitorName> **USER -scriptName** nsrdp.pl
- bind lb monitor <rdpMonitorName> <rdpServiceName>

Example

```
add service ser1 10.102.27.182 RDP 3389
add lb monitor RDP_MON USER -scriptName nsrdp.pl
bind lb monitor RDP_MON ser1
```

Parameter for configuring RDP scripting monitor

scriptName

Name of the script to be run

To configure a scripting monitor for RDP services by using the NetScaler configuration utility

1. In the **navigation pane**, expand **Load Balancing** and then click **Monitors**.
2. In the details pane, click **Add**.
3. In the **Create Monitor** dialog box, specify a name for the RDP monitor.
4. In the **Type** drop-down list, select **USER**.
5. On the **Special Parameters** tab, for the **Script Name**, click **Browse** and select nsrdp.pl from the default location.
6. Click **Create**.
7. From the **Monitors** pane, open the monitor you added, and verify the addition.
8. In the navigation pane, expand **Load Balancing**, and then click **Services.**
9. In the details pane, select the RDP service, and then click **Open**.
10. In the **Configure Service** dialog box, select the RDP scripting monitor that you added and click **Add.**
11. Click **OK**.

Use Cases

Certain deployment scenarios are useful to perform load balancing in a wide variety of circumstances. Several protocols benefit from a configuration that allows the server to respond directly to the client rather than through the NetScaler appliance. This is called *direct server return (DSR) mode*. Many deployments are faster when you connect the appliance to the network through a single interface, rather than placing it directly in the flow of traffic. This is called *one-arm mode*. Other deployments require that the appliance be installed in the flow of traffic, so that it transparently intercepts traffic that is sent to and from the servers that it manages. This is called *inline mode* or (occasionally) *two-arm mode*. These use cases are described in detail below.

Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream

Note: Load balancing virtual servers of type TCP and SSL_TCP support rule based persistence only on NetScaler 9.3.e.

Some protocols transmit name-value pairs in a TCP byte stream. The protocol in the TCP byte stream in this example is the Financial Information eXchange (FIX) protocol. In its traditional, non-XML implementation, the FIX protocol enables two hosts communicating over a network to exchange business or trade-related information as a list of name-value pairs (called "FIX fields"). The field format is <tag>=<value><delimiter>. This traditional tag-value format makes the FIX protocol ideal for the use case.

The tag in a FIX field is a numeric identifier that indicates the meaning of the field. For example, the tag 35 indicates the message type. The value after the equal sign holds a specific meaning for the given tag and is associated with a data type. For example, a value of A for the tag 35 indicates that the message is a logon message. The delimiter is the nonprinting "Start of Header" (SOH) ASCII character (0x01), which is the caret symbol (^). Each field is also assigned a name. For example, the field with tag 35 is the msgType field. Following is an example of a logon message.

```
8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52=20000426-12:05:06 98=0
108=30 10=157
```

Your choice of persistence type for a tag-value list such as the one shown above is determined by the options that are available to you for extracting a particular string from the list. Token-based persistence methods require you to specify the offset and length of the token that you want to extract from the payload. The FIX protocol does not allow you to do that, because the offset of a given field and the length of its value can vary from one message to another (depending on the message type, the preceding fields, and the lengths of the preceding values) and from one implementation to another (depending on whether custom fields have been defined). Such variations make it impossible to predict the exact offset of a given field or to specify the length of the value that is to be extracted as the token. In this case, therefore, rule based persistence is the preferred persistence type.

Assume that a virtual server fixlb1 is load balancing TCP connections to a farm of servers hosting instances of a FIX-enabled application, and that you want to configure persistence for connections on the basis of the value of the SenderCompID field, which identifies the firm sending the message. The tag for this FIX field is 49 (shown in the earlier logon message example).

To configure rule based persistence for the load balancing virtual server, set the persistence type for the load balancing virtual server to RULE and configure the rule parameter with an expression. The expression must be one that extracts the portion of the TCP payload in which you expect to find the SenderCompID field, typecasts the resulting string to a name-value list based on the delimiters, and then extracts the value of the SenderCompID field (tag 49), as follows:

```
set lb vserver fixlb1 -persistenceType RULE -rule
"CLIENT.TCP.PAYLOAD(300).TYPECAST_NVLIST_T('=','^').VALUE(\"49\")"
```

Note: Backslash characters have been used in the expression because this is a CLI command. If you are using the configuration utility, do not enter the backslash characters.

If the client sends a FIX message that contains the name-value list in the earlier logon message example, the expression extracts the value `INVMGR`, and the NetScaler appliance creates a persistence session based on this value.

The argument to the `PAYLOAD()` function can be as large as you deem is necessary to include the `SenderCompID` field in the string extracted by the function. Optionally, you can use the `SET_TEXT_MODE(IGNORECASE)` function if you want the appliance to ignore case when extracting the value of the field, and the `HASH` function to create a persistence session based on a hash of the extracted value. The following expression uses the `SET_TEXT_MODE(IGNORECASE)` and `HASH` functions:

```
CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=','^').SET_TEXT_MODE(IGNORECASE).VALUE
```

Following are more examples of rules that you can use to configure persistence for FIX connections (replace `<tag>` with the tag of the field whose value you want to extract):

- To extract the value of any FIX field in the first 300 bytes of the TCP payload, you can use the expressio `CLIENT.TCP.PAYLOAD(300).BEFORE_STR("^").AFTER_STR("<tag>=")`.
- To extract a string that is 20 bytes long at offset 80, cast the string to a name-value list, and then extrac the value of the field that you want, use the expression `CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPECAST_NVLIST_T('=','^').VALUE("<tag>"`
- To extract the first 100 bytes of the TCP payload, cast the string to a name-value list, and extract the value of the third occurrence of the field that you want, use the expression `CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T('=','^').VALUE("<tag>",2)`.

  Note: If the second argument that is passed to the `VALUE()` function is `n`, the appliance extracts the value of the $(n+1)^{th}$ instance of the field because the count starts from zero (`0`).

Following are more examples of rules that you can use to configure persistence. Only the payload-based expressions can evaluate data being transmitted through the FIX protocol. The other expressions are more general expressions for configuring persistence based on lower networking protocols.

- `CLIENT.TCP.PAYLOAD(100)`
- `CLIENT.TCP.PAYLOAD(100).HASH`
- `CLIENT.TCP.PAYLOAD(100).SUBSTR(5,10)`
- `CLIENT.TCP.SRCPORT`
- `CLIENT.TCP.DSTPORT`
- `CLIENT.IP.SRC`

- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

WAN Optimization – Load Balancing of Branch Repeater Appliances

WAN optimization can accelerate access to applications served from a distributed environment with remote branch office locations. Bandwidth management, Web caching, compression, data deduplication, data streamlining, Wide Area File Services (WAFS), HTTPS proxy, and Common Internet File System (CIFS) proxy are some techniques used in WAN optimization.

The Citrix® Branch Repeater™ product is a WAN optimization solution that accelerates desktop and application delivery, decreases WAN bandwidth consumption, and enables server consolidation. It is capable of handling real-time applications including VoIP and video streaming.

To provide support for high-scale data center deployment of WAN optimization, and improve the scalability and efficiency of the Branch Repeater appliances you can load balance the Branch Repeater appliances in the data centers and configure persistence.

Implementation of WAN Optimization

Load balancing of the Branch Repeater appliances for WAN optimization is implemented on nCore NetScaler appliances. WAN optimization is implemented as follows:

- A NetScaler appliance is deployed at the edge of a data center.
- You can accelerate all the traffic received by a virtual server or only specific traffic. Traffic to be accelerated is determined by the listen policies that are defined on the load balancing virtual servers. The policies can be based on aspects such as the VLAN or host from which the traffic originates or to which the traffic is destined.
- Configure a load balancing virtual server with a wildcard IP address and port (IP address as * and port as *) on the NetScaler to pick up the traffic.
- The Branch Repeater appliances are configured as services and bound to the load balancing virtual server.
- The L2 Connection option must be enabled on all the load balancing virtual servers involved in WAN optimization. When the L2 Connection option is enabled, the NetScaler identifies the connection with a 7-tuple of parameters: the normal 4-tuple (client IP address, client port, server IP address, and server port) and the three Layer 2 parameters (MAC address, VLAN ID, and channel number).

Deployment Scenarios

The optimal deployment of Branch Repeater appliances and the NetScaler appliances depends on the traffic acceleration requirements. Following are a few scenarios:

- Between a data center and many branch offices: A NetScaler appliance and Branch Repeater appliances are deployed in the data center, and one or more Branch Repeater appliances are deployed in each branch office. For more details, see WAN Optimization between Data Center and Branch Offices.
- Between a data center and clients, with each client using a Branch Repeater plug-in: The data center deployment includes a NetScaler appliance and Branch Repeater appliances. A Branch Repeater plug-in is installed on each client machine. For more details, see WAN Optimization between Clients and a Data Center.
- In a cloud: Many Branch Repeater virtual appliances are aggregated by a NetScaler appliance or a NetScaler VPX™ virtual appliance. This deployment helps to achieve elasticity and scaling. The Branch Repeater virtual appliances are typically integrated in a Citrix® XenDesktop® infrastructure. For more details, see WAN Optimization in a Cloud Scenario.

Settings for WAN Optimization

Load balancing of Branch Repeater appliances requires some specific settings on the Branch Repeater and NetScaler appliances.

Branch Repeater Setting

On each Branch Repeater appliance, you must enable the RETURN TO ETHERNET SENDER option.

NetScaler Settings

The following tables describe the settings necessary for a NetScaler appliance to load balance Branch Repeater appliances (or virtual appliances).

Table 1. Global Configuration

Parameter	Value
USIP	ON For all the services that represent Branch Repeater appliances
nsapimgr knob	nsapimgr -ys macmode_fwd_mypkt=1

Table 2. Virtual Server Configuration - Traffic through the Branch Repeater Appliances

Parameter	Value

IP address	*
Port	*
Service Type	ANY for the virtual server to load balance the Branch Repeater appliances TCP, HTTP, or any other service type for the virtual server to load balance the physical servers
MAC Based Forwarding	ON
L2 Connection	ON
Listen Policy	A listen policy to identify the traffic to be accelerated

Default Syntax Expressions

In addition to the listen policy expressions based on the parameters such as client IP address and port, server IP address and port, and TCP payload, you can define other listen policies for load balancing the Branch Repeater appliances. The following table lists the additional expressions that can be used.

Table 3. Default Syntax Expressions for WAN Optimization

Default syntax expression	Action
CLIENT.TCP.OPTIONS or SERVER. TCP.OPTIONS	Returns the TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.COUNT or SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.TYPE<integer> or SERVER.TCP.OPTIONS.TYPE<integer>	Returns the value of the option type. For each TCP option, an integer value is assigned.
CLIENT.TCP.OPTIONS.TYPE_NAME<enum> or SERVER.TCP.OPTIONS.TYPE_NAME<enum>	Returns the value of the specified option type present in the handshake packet sent by the client or server. The option type is specified by the value of enum. Valid values of enum are WANSCALER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG.
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns TRUE if the repeater option exists.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the IP address of the Branch Repeater present in the repeater option.
CLIENT.TCP.REPEATER_OPTION.MAC.<>	Returns the MAC address of the Branch Repeater present in the repeater option.

Persistence

For enhanced performance, you can configure persistence with Branch Repeater appliances just as you configure persistence with application servers. You can decide the persistence method according to the deployment scenario.

For example, in the deployment for WAN optimization between a data center and a client, the two load balancing virtual servers can be formed into a group and group persistence can be configured. In the deployment for WAN optimization in a cloud scenario or between a data center and branch offices, you can define your own rules to maintain persistence or use an existing persistence method.

For more information about persistence, see Configuring Persistence Based on User-Defined Rules and Configuring Group Persistence.

Parameters Applicable to WAN Optimization

Following is a list of load balancing methods, persistence methods, and monitor types supported for WAN optimization.

Load Balancing Methods

- ROUNDROBIN
- LEASTCONNECTION
- DESTINATIONIPHASH
- SOURCEIPHASH
- SRCIPDESTIPHASH
- LEASTBANDWIDTH
- LEASTPACKETS
- SRCIPSRCPORTHASH
- LRTM
- CUSTOMLOAD

Persistence Methods

- SOURCEIP
- DESTIP
- SRCIPDESTIP
- CUSTOMLOAD

- Rule-based persistence
- Group persistence

Monitor Types

- PING
- TCP

Configuring a NetScaler Appliance to Load Balance Branch Repeater Appliances

To configure a NetScaler appliance to load balance Branch Repeater appliances or virtual appliances, you must enable USIP mode, add the Branch Repeater appliances or virtual appliances as services, configure a virtual server with the L2 connection mode enabled and a listen policy specified, and bind the Branch Repeater services to the virtual server.

Note: If you want the NetScaler to bypass the Branch Repeater appliances if they are all DOWN, you can define the listen policy accordingly. The expression, `sys.vserver("<vServerName>").state.EQ(UP)`, makes sure that virtual server picks up the traffic only if it is UP. If a listen policy is already defined on the load balancing virtual server, type `&&` and append the above expression. If no listen policy has been defined, add a listen policy with the above expression. For more information, see WAN Optimization - Bypassing the Branch Repeater Appliances.

To configure the data-center NetScaler by using the NetScaler command line

At the NetScaler command prompt, type the following commands:

- enable ns mode USIP
- add service <serviceName> <serverName> <serviceType> <port>

  Repeat the above command as many times as necessary to add each Branch Repeater appliance as a service on the NetScaler.
- add lb vserver <name> <serviceType> <ip> <port> **-m** <redirectionMode> **-l2Conn** (ON| OFF) **-listenPolicy** <listen_policy_expression> **-listenPriority** <positive_integer>
- bind lb vserver <vserverName> <serviceName>
- shell nsapimgr **-ys** macmode_fwd_mypkt=1

  (Enables the nsapimgr knob.)

Example

```
> enable ns mode USIP MBF
 Done
> shell nsapimgr -ys macmode_fwd_mypkt=1
Changing macmode_fwd_mypkt from 0 to 1 ...  Done.
 Done
> add service SVC_BR_S1 192.0.2.6 ANY *
 Done
> add service SVC_BR_S2 192.0.2.7 ANY *
 Done
> add lb vserver LB_BR_S1 ANY * * -m MAC -l2conn ON  -listenpolicy CLIENT.TCP.OPTIONS.TYPE_NAME(REPEA
 Done
> bind lb vserver LB_BR_S1 SVC_BR_S1
 Done
> bind lb vserver LB_BR_S1 SVC_BR_S2
 Done
> add service SVC_TCP 192.0.2.8 TCP 3343
 Done
> add lb vserver v1 TCP 192.0.2.9 80 -l2conn ON -listenpolicy "CLIENT.TCP.OPTIONS.TYPE_NAME(REPEATER).
 Done
> bind lb vserver v1 SVC_TCP
 Done
>
```

Parameters for configuring a service

serviceName

Name of the service. This alphanumeric string is required. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Name or IP address of the server that is associated with the service. The IP address can be in either IPv4 or IPv6 format. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serviceType

Protocol used by the service.

port

Port on which the service listens.

Parameters for configuring a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

ip

IP address of the virtual server.

serviceType

Protocol used by the service.

port

Port on which the virtual server listens for client connections. The port number must be from 1 through 65534.

l2Conn

L2 connection mode. To identify the connection, the NetScaler uses the MAC, channel, and VLAN of the source machine, which are Layer 2 parameters, in addition to the client IP address, client port, source IP address, and source port. Possible values: ON, OFF. For load balancing of Branch Repeater appliances: ON.

listenPolicy

The policy expression that determines the traffic to which the virtual server listens.

For example, if you type the policy expression as `client.ip.src.eq(10.102.32.145)`, the virtual server listens to the traffic coming from the IP address 10.102.32.145.

m

The forwarding method.

To configure the data-center NetScaler by using the configuration utility

Perform the following tasks:

1. Enable **USIP** mode.
2. Enable the nsapimgr option.
3. Add all the Branch Repeater appliances as services.

4. Configure a load balancing virtual server with an appropriate listen policy, and with L2 connection enabled, and bind the Branch Repeater services to the virtual server.

5. Add a TCP service and a load balancing virtual server for TCP traffic.

To enable USIP mode

1. In the navigation pane, expand **System**, and then click **Settings**.

2. In the **Modes and Features** group, click **Configure Modes**.

3. In the **Configure Modes** dialog box, select the **Use Source IP** check box.

To enable the nsapimgr option

1. In the navigation pane, expand **System**, and then click **Diagnostics**.

2. In the **Utilities** group, click **Command Line Interface**.

3. In the **Command Line Interface** dialog box, type the following command in the **Command** box: `shell nsapimgr -ys macmode_fwd_mypkt=1`

To add the Branch Repeater appliances as services

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.

2. In the details pane, click **Add**.

3. Add each Branch Repeater as a service. In the **Create Service** dialog box, specify values for the following parameters, which correspond to the parameters described in "Parameters for configuring a service" as shown:

   - **Service Name***—serviceName
   - **Protocol***—serviceType
   - **Server***—serverName
   - **Port***—port

   *A required parameter

4. Click **Create**.

5. In the details pane, open the services you created and verify the settings.

To configure a load balancing virtual server for the Branch Repeater services

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which correspond to the parameters described in “Parameters for configuring a virtual server” as shown:
   - **Name*****—name
   - **Protocol*****—serviceType
   - **IP Address*****—ip
   - **Port*****—port

   *A required parameter
4. On the **Services** tab, select the services representing the Branch Repeater appliances that need to be load balanced.
5. On the **Advanced** tab:
   a. Select the **L2 Connection** check box.
   b. Click the **Listen Policy** link and type the listen policy expression for the virtual server. The policy should be to listen to the traffic from the Branch Repeater plug-in installed on a client or on a NetScaler at the branch office or at the other data center.
   c. In the **Redirection Mode** options, select **MAC Based**.
6. Click **Create**.
7. In the details pane, open the virtual server you created and verify the settings.

To add a TCP service

Follow the procedure described in "To add the Branch Repeater appliances as services," but select a TCP service instead of a Branch Repeater.

To add a load balancing virtual server for TCP traffic

Follow the procedure described in "To configure a load balancing virtual server for the Branch Repeater services."

You can add more virtual servers with specific listen policies to segregate different types of traffic.

WAN Optimization Between a Data Center and Branch Offices

For WAN optimization, a NetScaler appliance and multiple Branch Repeater appliances are deployed in the data center and Branch Repeater appliances are deployed in each branch office.

The following figure shows a typical topology for this type of deployment.

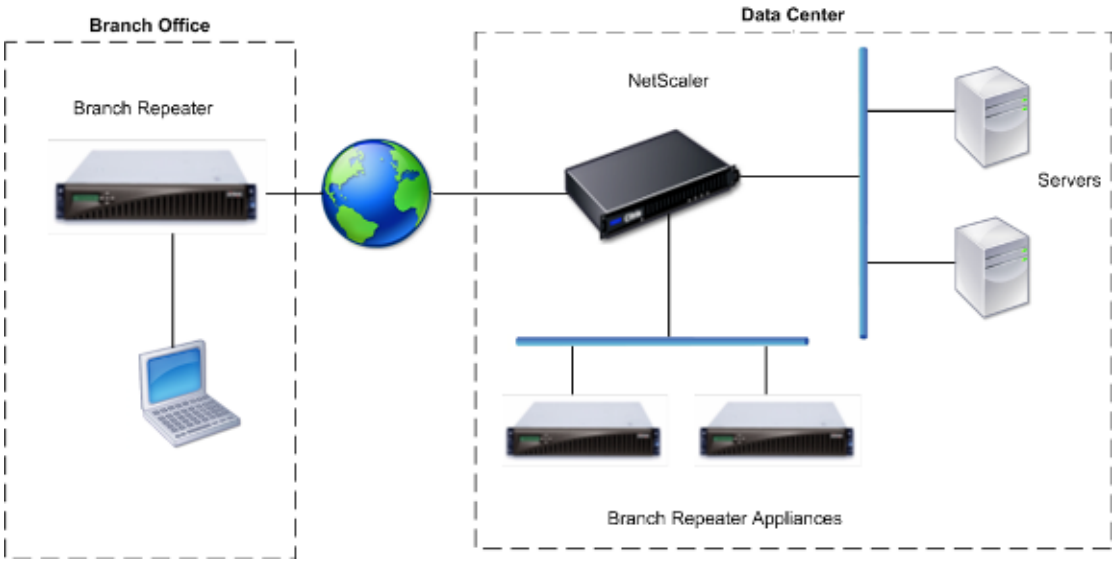


Figure 1. WAN Optimization between Data Center and Branch Offices

Traffic Flow

The traffic flows between a client and the data center as follows:

1. The client sends a request for a service.
2. The Branch Repeater plug-in processes the packet.
3. The ANY type virtual server on the NetScaler at the data center captures the request, as specified by the listen policy configured on this virtual server.
4. The NetScaler applies the load balancing method defined on the virtual server and forwards the request to the corresponding Branch Repeater appliance.
5. The Branch Repeater appliance processes the request and forwards it to the NetScaler.
6. As specified by the listen policy configured on the TCP type virtual server, this virtual server receives the request, applies the specified load balancing method, and sends it to the corresponding physical server for a response.

7. When the physical server sends a response to the client, the response returns by the same path.

The NetScaler uses the client IP address and forwards the traffic in a fully transparent mode. Make sure that the USIP option is enabled globally on the NetScaler appliance.

Settings for WAN Optimization

Load balancing of Branch Repeater appliances requires some specific settings on the Branch Repeater and NetScaler appliances.

Branch Repeater Setting

On each Branch Repeater appliance, you must enable the RETURN TO ETHERNET SENDER option.

NetScaler Settings

The following tables describe the settings necessary for a NetScaler appliance to load balance Branch Repeater appliances or virtual appliances.

Table 1. Global Configuration

Parameter	Value
USIP	ON For all the services that represent Branch Repeater appliances
nsapimgr knob	nsapimgr -ys macmode_fwd_mypkt=1

Table 2. Virtual Server Configuration — Traffic through the Branch Repeater Appliances

Parameter	Value
IP address	*
Port	*
Service Type	ANY for the virtual server to load balance the Branch Repeater appliances TCP, HTTP, or any other service type for the virtual server to load balance the physical servers
MAC Based Forwarding	ON
L2 Connection	ON
Listen Policy	A listen policy to identify the traffic to be accelerated

Default Syntax Expressions

In addition to the listen policy expressions based on the parameters such as client IP address and port, server IP address and port, and TCP payload, you can define other listen policies for load balancing the Branch Repeater appliances. The following table lists the additional expressions that can be used.

Table 3. Default Syntax Expressions for WAN Optimization

Default syntax expression	Action
CLIENT.TCP.OPTIONS or SERVER.TCP.OPTIONS	Returns the TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.COUNT or SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.TYPE<integer> or SERVER.TCP.OPTIONS.TYPE<integer>	Returns the value of the option type. For each TCP option, an integer value is assigned.
CLIENT.TCP.OPTIONS.TYPE_NAME<enum> or SERVER.TCP.OPTIONS.TYPE_NAME<enum>	Returns the value of the specified option type present in the handshake packet sent by the client or server. The option type is specified by the value of enum. Valid values of enum are WANSCALER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG.
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns TRUE if the repeater option exists.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the IP address of the Branch Repeater present in the repeater option.
CLIENT.TCP.REPEATER_OPTION.MAC.<>	Returns the MAC address of the Branch Repeater present in the repeater option.

Parameters Applicable to WAN Optimization

Following is a list of load balancing methods, persistence methods, and monitor types supported for WAN optimization.

Load balancing methods

- ROUNDROBIN
- LEASTCONNECTION
- DESTINATIONIPHASH

- SOURCEIPHASH
- SRCIPDESTIPHASH
- LEASTBANDWIDTH
- LEASTPACKETS
- SRCIPSRCPORTHASH
- LRTM
- CUSTOMLOAD

Persistence methods

- SOURCEIP
- DESTIP
- SRCIPDESTIP
- CUSTOMLOAD

Monitor types

- PING
- TCP

Sample Configuration

The following example shows how WAN optimization between a data center and branch offices can be configured on a NetScaler appliance.

Example

```
enable ns mode USIP MBF
add lb vs v1 any * * -m mac -l2conn on -listenpolicy client.tcp.options.type_name(repeater).exists
add service s1 BR1_IP any *
add service s2 BR2_IP any *
bind lb vs v1 s[1-2]
add expression e2 client.ether.srcmac.eq(BR1_MAC)||client.ether.srcmac.eq(BR2_MAC)
add lb vs v2 tcp IP1 80 -listenpolicy e2 –l2conn on
add service s3 SERVER1_IP tcp 80
```

```
add service s4 SERVER2_IP tcp 80
bind lb vs v2 s[3-4]
shell nsapimgr –ys macmode_fwd_mypkt =1
```

WAN Optimization Between Clients and a Data Center

For WAN optimization between a data center and clients, a NetScaler appliance and multiple Branch Repeater appliances are deployed in the data center, and a Branch Repeater Plug-in is installed on every client that communicates with the data center.

The following figure shows a typical topology for this type of deployment.

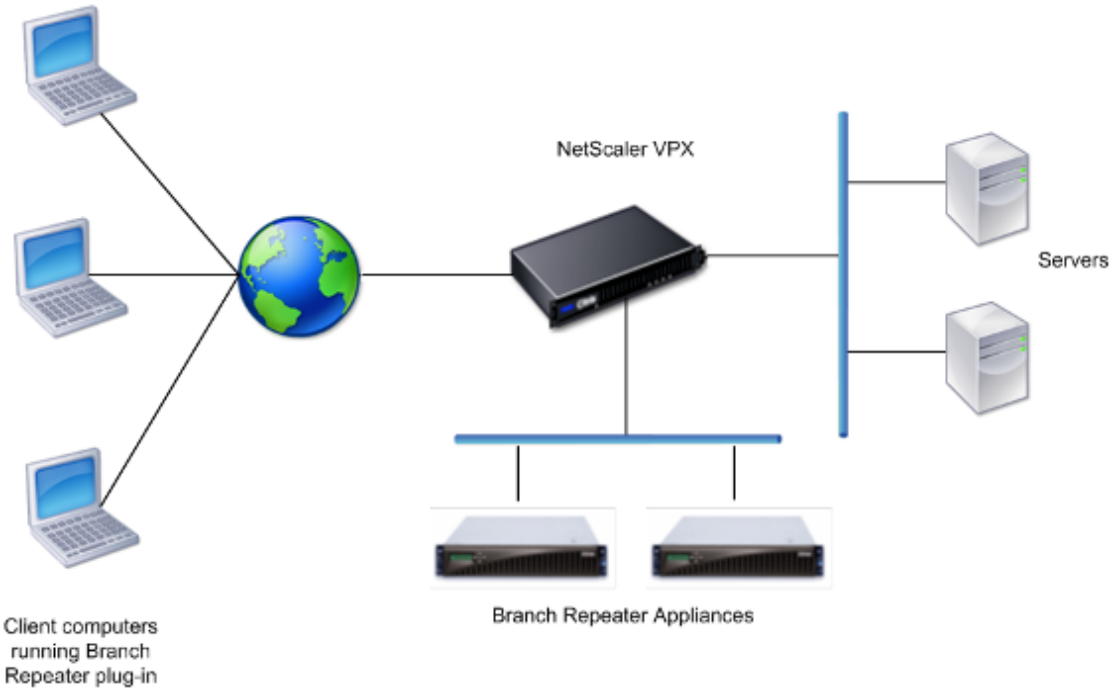


Figure 1. WAN Optimization between Data Center and Clients

Traffic Flow

Traffic flows between a client and the data center as follows:

1. The client sends a request for a service.
2. The ANY type virtual server on the NetScaler at the data center captures the request as specified by the listen policy configured on this virtual server.
3. The NetScaler applies the load balancing method defined on the virtual server and forwards the request to the corresponding Branch Repeater.
4. The Branch Repeater processes the request and forwards it to the NetScaler.

5. As specified by the listen policy configured on the TCP type virtual server, this virtual server receives the request, applies the specified load balancing method, and sends the request to the corresponding physical server for a response.

6. When the physical server sends a response to the client, the response returns by the same path.

The NetScaler uses the client IP address and forwards the traffic in a fully transparent mode. Make that the USIP option is enabled globally on the NetScaler appliance.

Settings for WAN Optimization

Load balancing of Branch Repeater appliances requires some specific settings on the Branch Repeater and NetScaler appliances.

Branch Repeater Setting

On each Branch Repeater appliance, you must enable the RETURN TO ETHERNET SENDER option.

NetScaler Settings

The following tables describe the settings necessary for a NetScaler appliance to load balance Branch Repeater appliances or virtual appliances.

Table 1. Global Configuration

Parameter	Value
USIP	ON For all the services that represent Branch Repeater appliances
nsapimgr knob	nsapimgr -ys macmode_fwd_mypkt=1

Table 2. Virtual Server Configuration — Traffic through the Branch Repeater Appliances

Parameter	Value
IP address	*
Port	*
Service Type	ANY for the virtual server to load balance the Branch Repeater appliances TCP, HTTP, or any other service type for the virtual server to load balance the physical servers
MAC Based Forwarding	ON
L2 Connection	ON

Listen Policy	A listen policy to identify the traffic to be accelerated

Default Syntax Expressions

In addition to the listen policy expressions based on the parameters such as client IP address and port, server IP address and port, and TCP payload, you can define other listen policies for load balancing the Branch Repeater appliances. The following table lists the additional expressions that can be used.

Table 3. Default Syntax Expressions for WAN Optimization

Default syntax expression	Action
CLIENT.TCP.OPTIONS or SERVER.TCP.OPTIONS	Returns the TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.COUNT or SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.TYPE<integer> or SERVER.TCP.OPTIONS.TYPE<integer>	Returns the value of the option type. For each TCP option, an integer value is assigned.
CLIENT.TCP.OPTIONS.TYPE_NAME<enum> or SERVER.TCP.OPTIONS.TYPE_NAME<enum>	Returns the value of the specified option type present in the handshake packet sent by the client or server. The option type is specified by the value of enum. Valid values of enum are WANSCALER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG.
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns TRUE if the repeater option exists.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the IP address of the Branch Repeater present in the repeater option.
CLIENT.TCP.REPEATER_OPTION.MAC.<>	Returns the MAC address of the Branch Repeater present in the repeater option.

Persistence

In deployment scenarios where a Branch Repeater Plug-in used, persistence should be maintained to ensure that the initial control connection and subsequent data connections from the same client go to the same Branch Repeater.

When a client sends a request, the ANY type load balancing virtual server on the NetScaler at the data center receives the request and forwards it to one of the Branch Repeater appliances according to the load balancing method specified for the load balancing virtual server. After the Branch Repeater processes the request, the TCP-type load balancing virtual server receives the request and forwards it to the physical application server according to the load balancing method specified for the TCP-type virtual server.

In such cases, you can group the two virtual servers on the NetScaler and define your own persistence rules, or use an existing persistence method, for the group. For more information about group persistence, see Configuring Persistence Groups.

To understand the configuration, see Sample Configuration.

Parameters Applicable to WAN Optimization

Following is a list of load balancing methods, persistence methods, and monitor types supported for WAN optimization.

Load balancing methods

- ROUNDROBIN
- LEASTCONNECTION
- DESTINATIONIPHASH
- SOURCEIPHASH
- SRCIPDESTIPHASH
- LEASTBANDWIDTH
- LEASTPACKETS
- SRCIPSRCPORTHASH
- LRTM
- CUSTOMLOAD

Persistence methods

- SOURCEIP
- DESTIP
- SRCIPDESTIP
- CUSTOMLOAD
- Rule-based persistence
- Group persistence

Monitor types

- PING
- TCP

Sample Configuration

Following is a sample Branch Repeater, NetScaler, and Branch Repeater Plug-in configuration for achieving WAN optimization between a data center and clients.

Configuration for the Branch Repeater Appliances

Following is the configuration on each Branch Repeater appliance in the data center. Instructions are provided for the Branch Repeater graphical user interface.

1. **Enable the `Send to Gateway` option.** The `Send to Gateway` option is enabled by default. However, if the option has been disabled, to enable it, on the **Configure Settings: Tuning** page, in **Virtual Inline**, click **Send to Gateway**.

2. **Configure a valid apA IP address.** The apA IP address must belong to the the apA network so that the NetScaler appliance can reach each of the Branch Repeater appliances directly. To specify an apA IP address, on the **Configure Settings: IP Address** page, in the **apA** section, in **IP Address**, enter the IP address that you want to use as the apA IP address.

3. **Set the gateway to the subnet IP address (SNIP) of the primary NetScaler appliance.** To set the gateway, on the **Configure Settings: IP Addresses** page, in the **apA** section, in **Gateway**, enter the SNIP of the primary NetScaler appliance.

4. **Enable the signaling channel for the Repeater Plug-in.** To enable the signaling channel for the Repeater Plug-in, on the **Configuration Settings: Repeater Plug-in** page, in the **Signaling Channel Configuration** tab, in **State**, click **Enabled**.

5. **Configure a valid signaling IP address.** To specify a signaling IP address, on the **Configure Settings: Repeater Plug-in** page, in **Signaling IP**, enter the signaling IP address.

6. **Specify the server subnets that need acceleration.** In **Configure Settings: Repeater Plug-in**, in the **Acceleration Rules** tab, specify the server subnets that need acceleration.

7. Replicate the configuration on the **Configure Settings: Repeater Plug-in** page across all the Branch Repeater appliances, except for the signaling IP address.

Configuration on the NetScaler Appliance

If you want to configure a NetScaler high availability (HA) pair, and you have not already configured one, run the `add ha node` command on both the NetScaler appliances. After you configure HA, check the status of both appliances. They should be up and enabled. Then, use the following instructions, which include sample commands, to configure WAN optimization. Run all the commands on the primary appliance in the HA pair.

1. Run the `shell nsapimgr -ys service_state_sync=0` command if you have configured HA, and then run the `shell nsapimgr -ys forward_icmp_fragments=1` command if you expect large ICMP packets. In an HA pair, if the secondary is UP when you set the `nsapimgr` knobs, the `nsapimgr` commands will be propagated to the secondary appliance. Otherwise, after you run the commands on the primary appliance, you must also run the commands on the secondary appliance. Update the `/nsconfig/nsafter.sh or /nsconfig/rc.netscaler` file with the `nsapimgr` commands so that the commands persist across reboots.

   ```
   > shell nsapimgr -ys service_state_sync=0
   Changing service_state_sync value from  1 to 0 ...  Done.
    Done
   > shell nsapimgr -ys forward_icmp_fragments=1
   Changing forward_icmp_fragments from 0 to 1 ...  Done.
    Done
   >
   ```

2. If the NetScaler appliance can reach the backend services directly, set the `preferDirectRoute` load balancing parameter to `NO`.

   ```
   > set lb parameter -preferDirectRoute NO
    Done
   >
   ```

3. Enable Use Source IP (USIP), L3, and MAC-based forwarding (MBF) modes.

   ```
   > enable ns mode USIP L3 MBF
    Done
   >
   ```

4. Enable the load balancing feature.

   ```
   > enable ns feature LB
    Done
   >
   ```

5. Configure the following Access Control Lists (ACLs) on the NetScaler appliance to allow only TCP traffic to be forwarded to the Branch Repeater appliances. After you configure these ACLs, all non-TCP traffic is bypassed.

   ```
   > add ns acl acl1 ALLOW -protocol TCP
    Done
   > add ns acl acl2 BRIDGE
   ```

```
 Done
> apply ns acls
 Done
>
```

6. Add a forwarding session to send traffic directly to a physical server if a virtual server is down or the listen policy does not match. You must add one forwarding session per server subnet.

```
> add forwardingsession session_bridge 192.0.2.10 255.255.255.0
 Done
>
```

7. Add a service for each data center Branch Repeater. The NetScaler appliance must be able to reach all the Branch Repeater appliances directly. The traffic sent from the NetScaler to a Branch Repeater appliance must not be intercepted by a router. In the following commands, two Branch Repeater appliances are represented by the services `DCBR1` and `DCBR2`.

```
> add service DCBR1 192.0.2.11 ANY * -usip YES
 Done
> add service DCBR2 192.0.2.12 ANY * -usip YES
 Done
>
```

   Note: If you add the services before you enable USIP mode globally on the NetScaler appliance, the services will not inherit the USIP option. You must then enable the USIP option on each service.

8. Configure the load balancing virtual servers.

```
> add lbvserver SigVIP ANY 192.0.2.13 * -listenpolicy '(SYS.VSERVER("SigVIP").STATE.EQ(UP) && CLIENT.T
 Done
> add lb vserver BR_LB ANY * * -Listenpolicy '(SYS.VSERVER("BR_LB").STATE.EQ(UP) && CLIENT.TCP.OPTI
 Done
>
```

9. Bind the load balancing virtual servers to the services that represent the Branch Repeater appliances.

```
> bind lb vserver BR_LB DCBR1
 Done
> bind lb vserver BR_LB DCBR2
 Done
> bind lbvserver SigVIP DCBR1
 Done
> bind lbvserver SigVIP DCBR2
 Done
>
```

10. Create a virtual server group that includes the virtual servers `SigVIP` and `BR_LB`, and then configure persistence for the group.

```
> bind lbgroup plugin SigVIP
 Done
```

```
> bind lbgroup plugin BR_LB
 Done
> set lbgroup plugin -persistencetype SOURCEIP
 Done
>
```

11. Save the configuration.

```
> save ns config
 Done
>
```

Configuration for the Branch Repeater Plug-In on a Client Machine

In the **Configuration** tab of the Branch Repeater Plug-in (Citrix Acceleration Manager), in the **Signalling Addresses** box, specify the signaling IP address. The signaling IP address is the IP address of the load balancing virtual server, `SigVIP`, that you configured on the Netscaler appliance. This should be delivered through Citrix Receiver or Merchandising server.

WAN Optimization in a Cloud Scenario

For WAN optimization in a cloud scenario, a NetScaler virtual appliance and multiple Branch Repeater virtual appliances are deployed in the data center, and a Branch Repeater plug-in is installed on every client that communicates with the data center.

The following figure shows a typical topology for this type of deployment.

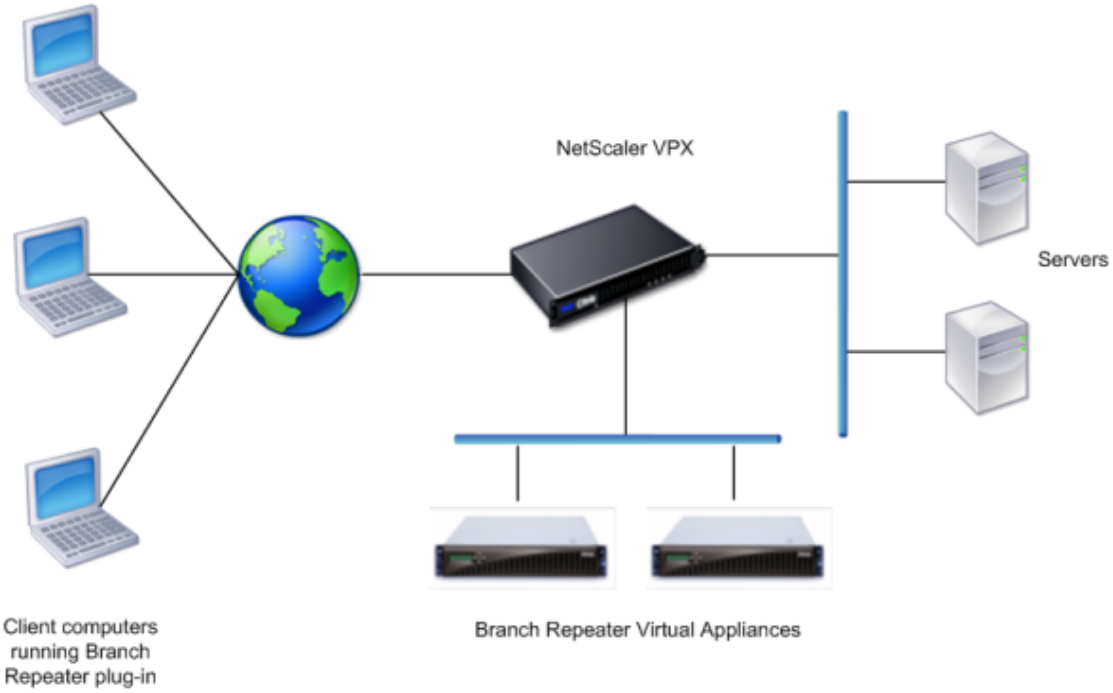


Figure 1. WAN Optimization in a Cloud Scenario

Traffic Flow

In a cloud scenario, traffic flow is as follows:

1. The client sends a request for a service.
2. The Branch Repeater plug-in installed on the client processes the request.
3. The NetScaler virtual appliance receives the request according to the listen policy defined on the virtual server.
4. The virtual NetScaler load balances the virtual Branch Repeater appliances and forwards the request to the virtual Branch Repeater selected by the specified load balancing method.

5. The Branch Repeater processes the request and sends it back to the NetScaler virtual appliance.
6. The NetScaler virtual appliance forwards the request to the physical server selected by the specified load balancing method.
7. The traffic from the physical server returns by the same path.

Settings for WAN Optimization

Load balancing of Branch Repeater appliances requires some specific settings on the Branch Repeater and NetScaler appliances.

Branch Repeater Setting

On each Branch Repeater appliance, you must enable the RETURN TO ETHERNET SENDER option.

NetScaler Settings

The following tables describe the settings necessary for a NetScaler appliance to load balance Branch Repeater appliances or virtual appliances.

Table 1. Global Configuration

Parameter	Value
USIP	ON For all the services that represent Branch Repeater appliances
nsapimgr knob	nsapimgr -ys macmode_fwd_mypkt=1

Table 2. Virtual Server Configuration — Traffic through the Branch Repeater Appliances

Parameter	Value
IP address	*
Port	*
Service Type	ANY for the virtual server to load balance the Branch Repeater appliances TCP, HTTP, or any other service type for the virtual server to load balance the physical servers
MAC Based Forwarding	ON
L2 Connection	ON
Listen Policy	A listen policy to identify the traffic to be accelerated

Default Syntax Expressions

In addition to the listen policy expressions based on the parameters such as client IP address and port, server IP address and port, and TCP payload, you can define other listen policies for load balancing the Branch Repeater appliances. The following table lists the additional expressions that can be used.

Table 3. Default Syntax Expressions for WAN Optimization

Default syntax expression	Action
CLIENT.TCP.OPTIONS or SERVER. TCP.OPTIONS	Returns the TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.COUNT or SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options present in the handshake packet sent by the client or server.
CLIENT.TCP.OPTIONS.TYPE<integer> or SERVER.TCP.OPTIONS.TYPE<integer>	Returns the value of the option type. For each TCP option, an integer value is assigned.
CLIENT.TCP.OPTIONS.TYPE_NAME<enum> or SERVER.TCP.OPTIONS.TYPE_NAME<enum>	Returns the value of the specified option type present in the handshake packet sent by the client or server. The option type is specified by the value of enum. Valid values of enum are WANSCALER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG.
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns TRUE if the repeater option exists.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the IP address of the Branch Repeater present in the repeater option.
CLIENT.TCP.REPEATER_OPTION.MAC.<>	Returns the MAC address of the Branch Repeater present in the repeater option.

Persistence

If NetScaler appliances or virtual appliances deployed in various branch offices are configured to load balance Branch Repeater appliances or plug-ins, you can maintain persistence between the Branch Repeater appliances or plug-ins at a branch office that is sending requests and the application server at the branch office that serves the requests.

To maintain persistence between the Branch Repeater appliances, you can define your own persistence rules or use an existing persistence method based on parameters such as source port, destination port, destination IP, interface, and VLAN. For example, you can define a persistence rule based on the MAC address of the Branch Repeater.

Note: When an AGEE plug-in is used, if you specify SOURCEIP persistence the load balancing of the Branch Repeater appliances may not be effective.

For more information about rule-based persistence, see Configuring Persistence Based on User-Defined Rules.

To understand the configuration, see Sample Configuration.

Parameters Applicable to WAN Optimization

Following is a list of load balancing methods, persistence methods, and monitor types supported for WAN optimization.

Load balancing methods

- ROUNDROBIN
- LEASTCONNECTION
- DESTINATIONIPHASH
- SOURCEIPHASH
- SRCIPDESTIPHASH
- LEASTBANDWIDTH
- LEASTPACKETS
- SRCIPSRCPORTHASH
- LRTM
- CUSTOMLOAD

Persistence methods

- SOURCEIP
- DESTIP
- SRCIPDESTIP
- CUSTOMLOAD
- Rule-based persistence
- Group persistence

Monitor types

- PING
- TCP

Sample Configuration

The following example shows a sample configuration on the NetScaler.

Example

```
enable ns mode USIP MBF
add lb vs v1 any * * -m mac -l2conn on -listenpolicy client.tcp.options.type_name(repeater).exists
add service s1 BR1_SIGNALLINGIP any *
add service s2 BR2_SIGNALLINGIP any *
bind lb vs v1 s[1-2]
add expression e2 client.ether.srcmac.eq(BR1_MAC)||client.ether.srcmac.eq(BR2_MAC)
add lb vs v2 tcp IP1 80 -listenpolicy e2 -l2conn on
add service s3 SERVER1_IP tcp 80
add service s4 SERVER2_IP tcp 80
bind lb vs v2 s[3-4]
add lb vs v3 any IP2 * -l2conn on
bind lb vs v3 s[1-2]
bind lb group grp1 v1
bind lb group grp1 v3
add expression e3 CLIENT.TCP.REPEATER_OPTION.IP
set lb group grp1 -persistencetype RULE -rule e3
shell nsapimgr –ys macmode_fwd_mypkt =1
```

Configure a signaling IP address on each Branch Repeater. Add the configured signaling IP address as the service for load balancing virtual servers V1 and V3.

In the Plug-in Manager, configure the IP address of the load balancing virtual server V3, which is configured on the NetScaler, as the signaling address for the Branch Repeater plug-in clients.

WAN Optimization - Bypassing the Branch Repeater Appliances

In a typical deployment scenario, a single data center is accessed by many branch offices with some of the branch offices having a Branch Repeater appliance to accelerate the traffic and some others without a Branch Repeater. In the data center, you can have the NetScaler appliance to load balance the Branch Repeater appliances and not load balance the physical servers in the server farm.

The requirements of such a scenario are:

- Traffic from a branch office that has a Branch Repeater should be directed to a Branch Repeater in the data center for acceleration before sending it to a physical server.
- Traffic from a branch office that does not have a Branch Repeater should be directly sent to a physical server. The Branch Repeater appliances should be bypassed.
- Non-TCP traffic from all branches, whether it comes through a Branch Repeater or not, should be directly sent to a physical server. The Branch Repeater appliances should be bypassed.
- Traffic from physical servers to clients should be accelerated only if it has come through a Branch Repeater.
- In a branch that has a Branch Repeater, traffic from certain users may need to be accelerated whereas traffic from some other clients need not be accelerated.
- If a Branch Repeater bound to a load balancing virtual server is DOWN or the listen policy does not match, then the traffic should go directly to a physical server.

The NetScaler implements the requirement in the following manner:

- Traffic to Branch Repeater appliances goes through a load balancing virtual server.
- The NetScaler uses listen policies on the load balancing virtual server to determine whether a certain request or response is to be accelerated or not. The Branch Repeater is bypassed if the traffic need not be accelerated.
- If a request from a specific client is accelerated, the response to the client is also accelerated.
- Forwarding sessions are created on the NetScaler to send traffic directly to a physical server if a virtual server is DOWN or the listen policy does not match.
- In case of non-TCP traffic, ACLs are used to bypass the Branch Repeater.
- To each load balancing virtual server, only one service is bound.

The following figure shows a sample deployment for this use case.

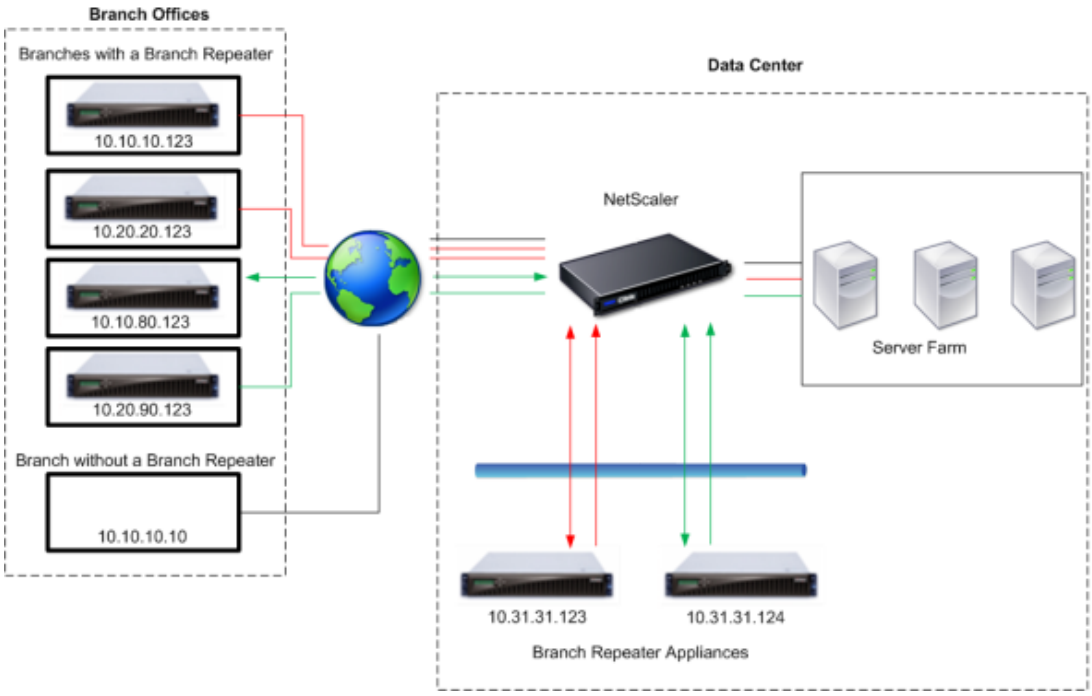


Figure 1. WAN Optimization when Branch Repeater Appliances are DOWN

Traffic Flow

Branch Office with a Branch Repeater - All Users

1. The user at the branch sends a request.
2. The Branch Repeater at the branch receives the request and forwards it to the NetScaler at the data center.
3. The load balancing virtual server on the NetScaler receives the request according to the listen policy. If the request is to be accelerated, NetScaler forwards the request to a Branch Repeater.
4. The Branch Repeater at the data center accelerates the request and forwards to the NetScaler.
5. The NetScaler forwards the request to a physical server in the server farm.
6. When the physical server sends the response, the NetScaler sends the response for acceleration to the same Branch Repeater.
7. The Branch Repeater accelerates the response and sends to the NetScaler.
8. The NetScaler sends to the response to the branch.
9. The Branch Repeater at the branch accelerates the response and sends to the client.

Branch Office with a Branch Repeater - Selected Users

1. The user at the branch sends a request.
2. The Branch Repeater at the branch accelerates the request.
3. The NetScaler receives the request and applies the listen policy.
   - If the request is from a user whose traffic need not be accelerated, the NetScaler forwards the request to a physical server in the server farm.
   - If the request is from a user whose traffic is to be accelerated, the NetScaler forwards the request to a Branch Repeater, and after receiving the accelerated request, forwards it to a physical server in the server farm.
4. When the physical server sends the response, if the response is to an accelerated request, the NetScaler forwards the response to a Branch Repeater and after receiving the accelerated response, forwards it to the branch office. If the response is to an unaccelerated request, it sends the response to the branch office.

Branch Office without a Branch Repeater

1. The user at the branch sends a request.
2. The NetScaler receives the request and forwards it to the physical server in the server farm.
3. When the physical server sends the response, the NetScaler sends the response to the client in branch office.

Settings for WAN Optimization

At the data center, the Branch Repeater appliances and NetScaler appliance require some specific settings. At branch offices, no specific settings are required on the Branch Repeater appliances.

Branch Repeater Settings

- Enable the **RETURN TO ETHERNET SENDER** option. (Use the menu sequence: **Configuration setting** > **Tuning** > **Virtual Inline**.)
- Configure a valid Auto Port Aggregation (APA) IP address. The Branch Repeater should be directly reachable by the NetScaler.
- Enable **SNMP Status**.
- In **Access Configuration**, it is recommended to set the **Community** string to `public`.
- Set **Management Station IP** to the NetScaler IP (NSIP).

NetScaler Settings

Go through the following steps:

1. Set the nsapimgr options.

2. Enable the load balancing feature, and enable USIP, L3, and MBF modes.
3. Add and apply ACLs to bypass non-TCP traffic.
4. Add a forwarding session for each server subnet.
5. Add the Branch Repeater appliances as services.
6. Create an SNMP monitor, and bind the SNMP and PING monitors to all the Branch Repeater services.
7. Add a load balancing virtual server for each Branch Repeater service and bind the service to the corresponding virtual server.

Note: For understanding the necessary setup, see Sample Configuration.

To configure the load balancing of Branch Repeater appliances by using the command line

To set the nsapimgr options

Type the following command from the shell prompt:

nsapimgr -ys skip_direct_rt_lkup=1

Execute the following commands from the NetScaler command prompt:

To enable the Load balancing feature, and the USIP, L3, and MBF modes

enable feature lb

enable ns mode usip l3 mbf

To add and apply ACLs on the NetScaler

add acl <aclname> ALLOW **-protocol** TCP

add acl <aclname> BRIDGE

apply acls

Note: Add the ACLs in the order mentioned above.

To add forwarding sessions

add forwardingsession <forwarding_session_name> <ipAddress_subnet_dc_server> <netmask>

Note: If the physical servers are in more than one subnet, add a forwarding session for each server subnet.

To add Branch Repeater appliances as services

Execute the following command for each Branch Repeater appliance:

add service <service_name> <ipAddress_BR> ANY *

Note: All the IP addresses (Branch Repeater appliances) should be reachable directly by the NetScaler without any router in the path.

To add and bind monitors to the Branch Repeater services

By default, when you create a Branch Repeater service, a ping monitor is created to monitor the health of the service. However, the ping monitor cannot detect whether a Branch Repeater is disabled or not. Therefore, configure an SNMP monitor providing the community name, Branch Repeater OID, and threshold, and bind the monitor to all the branch Repeater services.

add lb monitor <monitor_name> SNMP **-interval** <intervaltime> **-destPort** <destination_port> **-snmpOID** <snmp_oid> **-snmpCommunity** <snmp_community> **-snmpThreshold** <snmp_threshold_value>

…

bind lb monitor <snmp_monitor_name> < service_range>

bind lb monitor **ping** <service_range>

Note: Bind the PING monitor after binding the SNMP monitor.

To add load balancing virtual servers for the Branch Repeater services

Add one load balancing virtual server for each Branch Repeater service. Specify listen policies on the virtual server. In the listen policy, specify the branch whose traffic is to be accelerated by the NetScaler.

Case I. To accelerate all the traffic from a branch with a Branch Repeater, execute the following command for each virtual server to be added:

add lb vserver <virtualServerName> **ANY * * -m MAC -cltTimeout** <client_timeout_value> **-l2Conn ON -listenpolicy** (**SYS.VSERVER(**"<vserver_name>"**).STATE.EQ(UP) && ((CLIENT.TCP.OPTIONS.TYPE_NAME(REPEATER).EXISTS && CLIENT.TCP.REPEATER_OPTION.IP.EQ(**<IP_Address_ofBranchOffice_BR>**)) || CLIENT.IP.DST.IN_SUBNET(**<branch subnet>**))) -listenpriority** <listen_priority_value>

bind lb vserver <vServerName> <serviceName>

Case II. To accelerate traffic of only selected users from a branch with a Branch Repeater, execute the following command for each virtual server to be added:

add lb vserver <virtualServerName> ANY * * **-m** MAC **-cltTimeout** <client_timeout_value> **-l2Conn** ON **-listenpolicy (SYS.VSERVER(**"<vserver_name>"**).STATE.EQ(UP) && (CLIENT.IP.SRC.EQ**<client1_IP>**) || CLIENT.IP.DST.EQ(**<client2_IP>**))) -listenpriority** <listen_priority_value>

bind lb vserver <vServerName> <serviceName>

Note: In the client.ip.dst.in_subnet, for the branch_subnet value, you can specify more than one branch subnet.

To configure the load balancing of Branch Repeater appliances by using the NetScaler configuration utility

To enable the NetScaler feature and modes

In the navigation pane, click **System**, and then click **Settings**.

- To enable the load balancing feature:
  1. In the details pane, click **Configure basic features**.
  2. In the **Configure basic features** dialog box, select the **Load Balancing** check box.
  3. Click **OK**.
- To enable the modes:
  1. In the details pane, click **Configure modes**.
  2. In the **Configure Modes** dialog box, select the following check boxes:
     - **Use Source IP**
     - **MAC Based Forwarding**
     - **Layer 3 Mode (IP Forwarding)**
  3. Click **OK**.

To add and apply ACLs on the NetScaler

1. In the navigation pane, click **Network**, and then click **ACLs**.
2. In the details pane, click the **Extended ACLs** tab, and then click **Add**.
3. In the **Name** box, type a name for the ACL.
4. In the **Action** dropdown list, select **ALLOW**.
5. In the **Protocol** dropdown list, select **TCP**.
6. Click **Create**, and then **Close**.
7. Select the ACL you created and click **Apply**.

Add another ACL by repeating the above steps, except that in Step 4, select **BRIDGE** in the **Action** dropdown list, and omit Step 5.

Note: Add the ACLs in the order mentioned above.

To add forwarding sessions

1. In the navigation pane, click **Network**, and then click **Forwarding Sessions**.
2. In the details pane, click **Add.**
3. In the **Create Forwarding Sessions** dialog box, in the **Name** box, type a name for the forwarding session.
4. Select the **Subnet** box.
5. In the **Subnet IP** box and the **Netmask** box, specify values.
6. Click **Create**, and then **Close**.

To add Branch Repeater services

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, set values for the following parameters:
   - **Service Name**
   - **Server**
   - **Protocol** should be ANY
   - **Port** should be *
4. Click the **Advanced** tab, and in the **Idle Time-out** box, specify values for the client timeout and server timeout.
5. Click **Create**, and then **Close**.

Repeat the following procedure for each Branch Repeater.

Note: All the IP addresses (Branch Repeater appliances) should be reachable directly by the NetScaler without any router in the path.

To add and bind monitors to the Branch Repeater services

By default, when you create a Branch Repeater service, a ping monitor is created to monitor the health of the service. However, the ping monitor cannot detect whether a Branch Repeater is disabled or not. Therefore, configure an SNMP monitor providing the community name, Branch Repeater OID, and threshold, and bind the monitor to all the branch Repeater services.

To create an SNMP monitor:

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. In the details pane, click **Add**.

3. In the **Create Monitor** dialog box, set values for the following parameters:
    - **Name**
    - **Type** should be **SNMP**
4. Click the **Standard Parameters** tab, and set values for the following parameters:
    - **Interval**
    - **Destination Port**
5. Click the **Special Parameters** tab, and set values for the following parameters:
    - **SNMP Community**
    - **SNMP OID**
    - **SNMP Threshold**
6. Click **Create**, and then **Close**.

To bind the SNMP monitor you created and the default PING monitor to the Branch Repeater service:

1. In the navigation pane, expand **Load Balancing**, and then click **Services.**
2. In the details pane, select the Branch Repeater service, and then click **Open.**
3. In the **Configure Service** dialog box, click the **Monitors** tab.
4. In the **Available** list, select the SNMP monitor you created and click **Add**. The SNMP monitor name appears in the list of **Configured monitors**.
5. Click **OK**, and then **Close.**
6. After binding the SNMP monitor, bind the default PING monitor by using the same procedure.

Repeat Steps 1 to 6 for each Branch Repeater service.

To add load balancing virtual servers for the Branch Repeater services

Add one load balancing virtual server for each Branch Repeater service. Specify the listen policy on the virtual server so that the virtual server bypasses the Branch Repeater for TCP traffic not to be accelerated.

Case I

To accelerate all the traffic from a branch with a Branch Repeater, add a virtual server by using the following procedure:

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, set values for the following parameters:

- **Name**
- **IP Address** should be *
- **Protocol** should be ANY
- **Port** should be *

4. Click the **Services** tab, and select the check box corresponding to the Branch Repeater service for which you are adding the virtual server.
5. Click the **Advanced** tab, and set values for the following parameters:
   - **Redirection Mode** should be **MAC Based** .
   - **L2 Connection** box should be checked.
6. Click the **Advanced** tab, expand **Listen Policy**, and specify the following values:
   - **Listen Priority**
   - In the **Listen Policy** rule, type the following rule:

     ```
     (SYS.VSERVER("<vserver_name>").STATE.EQ(UP) &&
     ((CLIENT.TCP.OPTIONS.TYPE_NAME(REPEATER).EXISTS &&
     CLIENT.TCP.REPEATER_OPTION.IP.EQ(<IP_Address_ofBranchOffice_BR>))
     || CLIENT.IP.DST.IN_SUBNET(<branch subnet>)))
     ```

     Specify the actual values for virtual server name, IP address of the branch office BR, and branch subnet.
7. Click **Create**, and then **Close**.

Repeat the above procedure for each Branch Repeater.

Case II

To accelerate traffic of only selected users from a branch with a Branch Repeater, add a virtual server by using the above procedure and set the listen policy as shown below:

```
SYS.VSERVER("<vserver_name>").STATE.EQ(UP) &&
(CLIENT.IP.SRC.EQ(client1 IP) || CLIENT.IP.DST.EQ(clinet2 IP))
```

Sample Configuration

```
## Set nsapimgr knobs
 nsapimgr -ys skip_direct_rt_lkup=1
 ## Enable lb
 enable feature LB
## Enable modes
enable ns mode USIP L3 MBF
## Configure ACLs to make sure that only TCP traffic is sent to BR and all the non-TCP traffic is bypassed.
add acl acl1 ALLOW -protocol TCP
add acl acl2 BRIDGE
apply acls
```

```
## Add forwarding sessions for each server subnet
add forwardingsession fwsess1 10.12.12.214 255.255.255.255
add forwardingsession fwsess2 10.12.12.215 255.255.255.255
## Add BR services. The IP address is APA IP address
add service brs1 10.31.31.123 ANY *
add service brs2 10.31.31.124 ANY *
## Add an SNMP monitor and bind to BR services
add lb monitor monitor1 SNMP -interval 20 -destPort 161 -snmpOID 1.3.6.1.4.1.3845.30.4.1.1.1.1.0 -snmpCo
bind lb monitor monitor1 s[1-2]
bind lb monitor PING s[1-2]
##Add lb vservers
add lb vserver v1 ANY * * -m MAC -l2Conn ON -listenpolicy (SYS.VSERVER("vsname").STATE.EQ(UP) && ((CLIEN
CLIENT.IP.DST.IN_SUBNET(10.10.10.1))) -listenpriority 1
##To accelerate traffic from only selected users
add lb vserver v2 ANY * * -m MAC -l2Conn ON -listenpolicy SYS.VSERVER("v10").STATE.EQ(UP) && (CLIENT.IP.S
CLIENT.IP.DST.IN_SUBNET(10.20.20.1))) -listenpriority 2
bind lb vserver v1 brs1
bind lb vserver v2 brs2
## Save the configuration
save ns config
```

Configuring Load Balancing in Direct Server Return Mode

Load balancing in direct server return (DSR) mode allows the server to respond to clients directly using a return path that does not flow through the NetScaler appliance. In DSR mode, however, the appliance can continue to perform health checks on services. In a high-data volume environment, sending server traffic directly to the client in DSR mode increases the overall packet handling capacity of the appliance because the packets do not flow through the appliance.

DSR mode has the following features and limitations:

- It supports one-arm mode and inline mode.
- The appliance ages out sessions based on idle timeout.
- Because the appliance does not proxy TCP connections (that is it does not send SYN-ACK to the client), it does not completely shut out SYN attacks. By using the SYN packet rate filter, you can control the rate of SYNs to the server. To control the rate of SYNs, set a threshold for the rate of SYNs. To get protection from SYN attacks, you must configure the appliance to proxy TCP connections. However, that requires the reverse traffic to flow through the appliance.

In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service, and the service responds to clients directly, bypassing the NetScaler. The following table lists the names and values of the entities configured on the NetScaler in DSR mode.

Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
Monitors	TCP	None	None

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

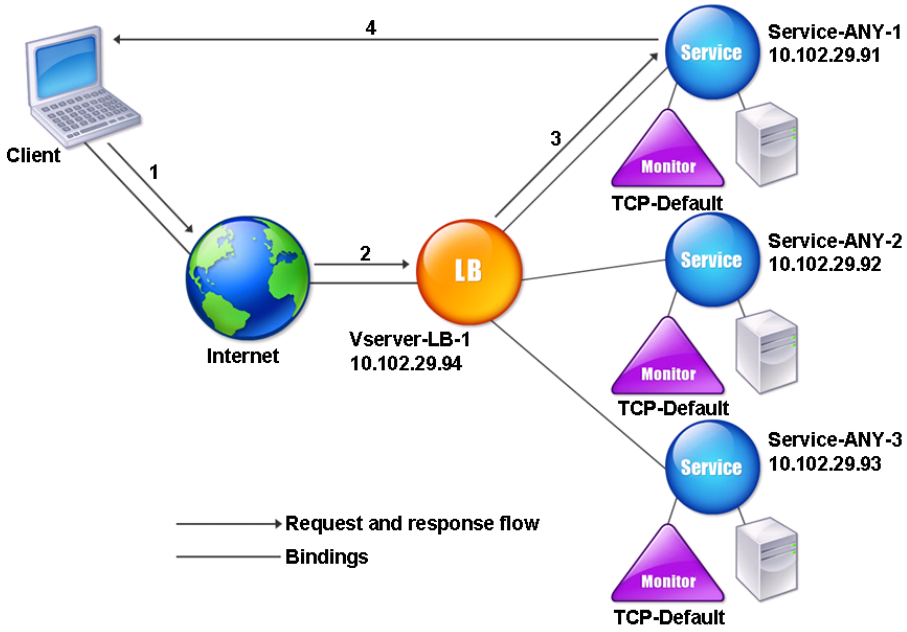


Figure 1. Entity Model for Load Balancing in DSR Model

For the appliance to function correctly in DSR mode, the destination IP in the client request must be unchanged. Instead, the appliance changes the destination MAC to that of the selected server. This setting enables the server to determine the client MAC address for forwarding requests to the client while bypassing the server. To enable the appliance to do this, you must enable MAC-based forwarding.

To enable MAC-based forwarding by using the NetScaler command line

At the NetScaler command prompt, type:

enable ns mode MAC

To enable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. On the **Settings** pane, under **Modes and Features**, click **Configure modes**.
3. In the **Configure Modes** dialog box, select the **MAC Based Forwarding** check box, and then click **OK**.
4. In the **Enable/Disable Mode(s)?** dialog box, click **Yes**.

Next, you configure a basic load balancing setup as described in Setting Up Basic Load Balancing, naming the entities and setting the parameters using the values described in the previous table.

After you configure the basic load balancing setup, you must customize it for DSR mode. To do this, you configure a supported load balancing method, such as the Source IP Hash method with a sessionless virtual server. You also need to set the redirection mode to allow the server to determine the client MAC address for forwarding responses and bypass the appliance.

After you configure the load balancing method and redirection mode, you need to enable the USIP mode on each service. The service then uses the source IP address when forwarding responses.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**lbMethod** <LBMethodOption> -**m** <RedirectionMode> -**sessionless** <Value>

Example

set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless enabled

To configure the load balancing method and redirection mode for a sessionless virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server (for example, **Vserver-LB-1**), and then click **Open**.
3. On the **Method and Persistence** tab, under **LB Method**, select **SOURCE IP Hash**.
4. On the **Advanced** tab, under **Redirection Mode**, select the **MAC Based**.
5. Select the **Sessionless** check box and click **OK**.

To configure a service to use source IP address by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**usip** <Value>

Example

set service Service-ANY-1 -usip yes

To configure a service to use source IP address by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. On the **Services** pane, click **Service-ANY-1**, and then click **Open**.

3. On the **Advanced** tab, under **Settings**, select the **Use Source IP** check box, and then click **OK**.

4. Repeat steps 1-5 for the services **Service-ANY-2** and **Service-ANY-3**.

Note: For USIP to function correctly, you must set it globally. For more information about configuring USIP globally, see the "IP Addressing" chapter in the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671.

Certain additional steps are required in certain situations, which are described in the succeeding sections.

Configuring LINUX Servers in DSR Mode

The LINUX operating system requires that you set up a loopback interface with the NetScaler appliance virtual IP address (VIP) on each load balanced server in the DSR cluster.

To configure LINUX server in DSR mode

To create a loop back interface with the NetScaler appliance's VIP on each load balanced server, at the Linux OS prompt type the following commands:

```
ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
```

Then, run the software that re-maps the TOS id to VIP.

Note: Add the correct mappings to the software before running it. In the preceding commands, the LINUX server uses dummy0 to connect to the network. When you use this command, type the name of the interface that your LINUX server uses to connect to the network.

Configuring DSR Mode When Using TOS

Differentiated services (DS), also known as TOS (Type of Service), is a field that is part of the TCP packet header. TOS is used by upper layer protocols for optimizing the path for a packet. The TOS information encodes the NetScaler appliance virtual IP address (VIP), and the load balanced servers extract the VIP from it.

In the following scenario, the appliance adds the VIP to the TOS field in the packet and then forwards the packet to the load balanced server. The load balanced server then responds directly to the client, bypassing the appliance, as illustrated in the following diagram.

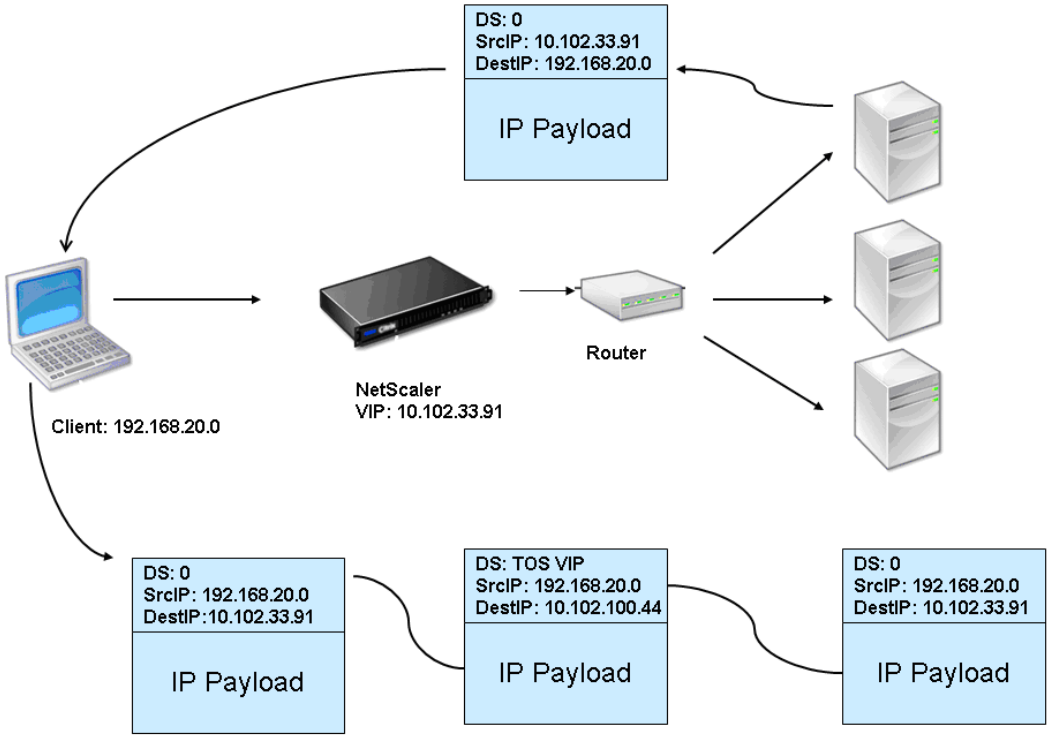


Figure 1. The NetScaler Appliance in DSR mode with TOS

The TOS feature is specifically customized for a controlled environment, as described below:

- The environment must not have any stateful devices, such as stateful firewall and TCP gateways, in the path between the appliance and the load balanced servers.
- Routers at all the entry points to the network must remove the TOS field from all incoming packets to make sure that the load balanced server does not confuse another TOS field with that added by the appliance.

- Each server can have only 63 VIPs.
- The intermediate router must not send out ICMP error messages regarding fragmentation. The client will not understand the message, as the source IP address will be the IP address of the load balanced server and not the NetScaler VIP.
- TOS is valid only for IP-based services. You cannot use domain name based services with TOS.

In the example, Service-ANY-1 is created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to the service, and the service responds to clients directly, bypassing the appliance. The following table lists the names and values of the entities configured on the appliance in DSR mode.

Entity Type	Name	IP Address	Protocol
Virtual server	Vserver-LB-1	10.102.33.91	ANY
Services	Service-ANY-1	10.102.100.44	ANY
Monitors	PING	None	None

DSR with TOS requires that load balancing be set up on layer 3. To configure a basic load balancing setup for Layer 3, see Setting Up Basic Load Balancing. Name the entities and set the parameters using the values described in the previous table.

After you configure the load balancing setup, you must customize the load balancing setup for DSR mode by configuring the redirection mode to allow the server to decapsulate the data packet and then respond directly to the client and bypass the appliance.

After specifying the redirection mode, you can optionally enable the appliance to transparently monitor the server. This enables the appliance to transparently monitor the load balanced servers.

To configure the redirection mode for the virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**m** <Value> -**tosId** <Value>

Example

set lb vserver Vserver-LB-1 -m TOS -tosId 3

To configure the redirection mode for the virtual server by using the configuration utility

1. In the left navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** pane, select the virtual server (for example, **Vserver-LB-1**) and click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Advanced** tab, in **Redirection Mode**, click **TOS Based**.
4. In the **TOS Id** box, enter a value for the **TOS ID**, (for example, **3**).
5. Click **OK**.

To configure the transparent monitor for TOS by using the NetScaler command line

At the NetScaler command prompt, type:

add monitor <MonitorName> <Type> -**destip** <DestinationIP> -**tos** <Value> -**tosId** <Value>

Example

add monitor mon1 PING -destip 10.102.33.91 -tos Yes -tosId 3

To create the transparent monitor for TOS by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Monitors**.
2. On the **Monitors** pane, select the monitor (for example, **tcp**), and click **Add**.
3. In the **Create Monitor** dialog box, in the **Name** and **Destination IP** boxes, enter the monitor name and the destination IP address (for example, **PING** and **10.102.33.91**).
4. In the **Type** list, select the type of monitor (for example, **PING**).
5. To configure the monitor for TOS, select the **TOS** check box.
6. In the **TOS Id** box, enter the same TOS ID that you had entered for the virtual server (for example, **3**.)
7. Click **OK**.

Configuring Load Balancing in DSR Mode by Using IP Over IP

You can also configure your NetScaler appliance to use direct server return (DSR) mode across Layer 3 networks by using IP tunneling, also called *IP over IP* configuration. As with standard load balancing configurations for DSR mode, this allows protected servers to respond to clients directly instead of using a return path through the NetScaler appliance, improving response times and throughput. As with standard DSR mode, the NetScaler appliance monitors the protected servers and performs health checks on the application ports.

With IP over IP configuration, the NetScaler appliance and the servers that it protects do not need to be on the same Layer 2 subnet. Instead, the NetScaler appliance accepts requests on the Mapped IP (MIP) assigned to the protected server, and then superencrypts the packets before resending them to the protected destination server. After the destination server receives the packets, it removes the superencryption, and then sends its responses directly to the client.

To configure IP over IP DSR mode on your NetScaler appliance, you must do the following:

- **Enable MAC-based forwarding.**
- **Create services.** Create a service for each of your protected back-end applications. Assign a mapped IP and at least one appropriate monitor to each of them.
- **Create a load balancing virtual server.** Create a virtual server, assigning a mapped IP to it. Set the Protocol to Any, the Port to *, and configure it to be stateless. Bind the services that you created to the virtual server.
- **Enable IP tunnel based redirection.** Enable IP tunneling based redirection for this virtual server.

Note: The instructions that follow assume familiarity with basic NetScaler load balancing or content switching configuration. If you are not familiar with configuring the NetScaler appliance, you should review the first three sections of this chapter and Configuring Load Balancing in Direct Server Return Mode, before attempting to configure DSR mode using IP over IP.

Enabling MAC-Based Forwarding

You must first enable MAC-based forwarding to configure DSR mode. In DSR mode, the NetScaler appliance does not change the destination IP in the client request, but modifies the destination MAC to the MAC address of the selected load-balanced server

To enable MAC-based forwarding by using the NetScaler command line

At the NetScaler command prompt, type:

enable ns mode <ConfigureMode>

Example

enable ns mode MAC

To enable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. On the **Settings** pane, under **Modes and Features**, click **Change modes**.
3. In the **Configure Modes** dialog box, select the **MAC Based Forwarding** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, and then click **Yes**.

Configuring Services for IP over IP DSR

After enabling MAC-based forwarding, you must next configure one service for each of your protected applications. The service handles traffic from the NetScaler appliance to those applications, and allows the NetScaler appliance to monitor the health of each protected application.

You assign a service type of **ANY** and a port of * to your new service, and configure it for USIP mode. You can also bind a monitor to the service if you want the NetScaler appliance to monitor the health of the application.

Note: If you are unfamiliar with the general process of creating services, you can review Configuring Services.

To create and configure a service for IP over IP DSR by using the NetScaler command line

At the NetScaler command prompt, type the following commands in this order:

- add service <serviceName> <serverName> <serviceType> <port> -**usip** <usip>
- add monitor <monitorName> <monitorType> -**destip** <ip> -**iptunnel** <iptunnel>

Example

add service Service-DSR-1 10.102.29.5 ANY * -usip yes

add monitor mon-1 PING -destip 10.102.33.91 -iptunnel yes

Service Configuration Parameters

serviceName

Name of the service that you are creating. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

You cannot change a service name after you create the service.

serverName

IP address of the server that is associated with the service. The IP address can be in either IPv4 or IPv6 format.

When you provide an IP address, a server object is created with this IP address as its name. If the server is not reachable from the NetScaler or is not active, the service state is shown as DOWN.

serviceType

Behavior of the service. Select one of the following service types: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. For more information about each service type, see its description under Configuring Services.

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

usip

Set to YES for IP over IP DSR, to configure the virtual server to use source IP mode.

monitorName

The name of the monitor that you are binding to the service. For more information about the different types of monitors and how to configure them, see Monitors.

monitorType

The type of monitor that you are binding to the service.

destip

The IP of the service or server object for the protected application that the monitor watches.

iptunnel

Set to YES for IP over IP DSR, to configure the virtual server to use IP tunneling.

To create and configure a service for IP over IP DSR by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, do one of the following:
   - To create a new service, click **Add**.
   - To modify an existing service, select the service, and then click **Open**.
3. In the **Create Service** dialog box, specify values for the following parameters, which correspond to parameters described in "Service Configuration Parameters" as shown:
   - Service Name*—name
   - Protocol*—type
   - Server*—IP
   - Port*—port

   * A required parameter
4. Click **Create**, and then click **Close**.

Configuring a Load Balancing Virtual Server

After creating services for your load-balanced servers, you next configure a virtual server to handle requests to your protected applications.

You assign a service type of **ANY** and a port of * to your virtual server. You can configure any load balancing method that you want to use. You set the forwarding method to IPTUNNEL, and configure the virtual server to operate in sessionless mode.

If you are unfamiliar with the general process of creating virtual servers, see Creating a Virtual Server.

To create and configure a load balancing virtual server for IP over IP DSR by using the NetScaler command line

At the NetScaler command prompt type the following command:

add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <port> -**lbMethod** <method> -**m** <ipTunnelTag> -**sessionless** <sessionless>

Example

add lb vserver Vserver-LB-1 ANY 10.102.29.60 * -lbMethod SourceIPHash -m IPTUNNEL -sessionless enabled

To bind a service to a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt type the following command:

bind lb vserver <name> <serviceName>

Example

bind lb vserver Vserver-LB-1 Service-DSR-1

To verify the configuration of a load balancing virtual server by using the NetScaler command line

At the NetScaler command prompt type the following command:

show lb vserver <name>

Example

show lb vserver Vserver-LB-1

Parameters for configuring virtual servers

name

A name for your new virtual server. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

Protocol

The protocol that your virtual server processes. For an IP over IP DSR virtual server, set the protocol to ALL.

IP

The IP address assigned to your virtual server. This is normally an Internet-routable IP.

Note: If the virtual server uses IPv6, select the IPv6 check box and enter the address in IPv6 format. (An IPv6 format address appears as follows: 1000:0000:0000:0000:0005:0600:700a:888b.)

Port

The port that your virtual server listens on for traffic. For an IP over IP DSR virtual server, set the port to *.

Method

The load balancing method to use for this load balancing configuration. For information on the various load balancing types, see Load Balancing Algorithms.

ipTunnelTag

Enables IP tunneling. For an IP over IP DSR virtual server, set to IPTUNNEL.

sessionless

When set to sessionless, configures the virtual server to operate in sessionless mode.

To create and configure a load balancing virtual server for IP over IP DSR by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring virtual servers" as shown:
   - Name*—name
   - Protocol*—protocol
   - IP address*—IPAddress
   - Port*—port

   * A required parameter
4. In the **Services** tab, select the check box beside the name of each service that routes traffic to a server in your load balancing setup, and do any additional configuration that is necessary.
   - You can adjust the priority column by using the spin buttons to the right of the priority number assigned to the service.
   - You can modify the settings for any service by selecting it, and then clicking **Open** to open the **Configure Service** dialog box for that service.
   - If you have not already created a service for each of your load balancing servers, you can click **Add** to open the **Create Servic**e dialog box and add a service.
5. In the **Advanced** tab, under **Redirection Mode**, select **IP Tunnel Based**.
6. Click **Create**, and then click **Close**. The virtual server that you created now appears in the **Virtual Servers** pane.

Enabling IP Tunnel-Based Redirection

Finally, you add the IP tunnels needed so that the NetScaler can route tunneled packets properly. You add one tunnel for each service that accepts traffic to your load balanced servers, which requires that you configure both the entry point of the IP tunnel on NetScaler applianceand the exit point on each of the load-balanced servers in the load balancing group. This means that you must issue the commands below once for each service.

To create and configure an IP tunnel for IP over IP DSR by using the NetScaler command line

At the NetScaler command prompt type the following command:

- add iptunnel <name> <remoteIp> <remoteSubnetMask> <localIp>
- show iptunnel <name>
- add route <subnet> <name>
- show route <subnet>

Example

add iptunnel lb-dsr-tunnel-1 10.102.40.123 255.255.255.255 10.56.223.81

show iptunnel lb-dsr-tunnel-1

add route 10.102.40.123 255.255.255.255 lb-dsr-tunnel-1

show route 10.102.40.123 255.255.255.255

To create and configure an IP tunnel on the load balanced server for IP over IP DSR

Log on to the load balanced server, and at the UNIX shell prompt type the following command:

- add iptunnel tun1 <remoteIp> <remoteSubnetMask> <localIp>/*

- show iptunnel <name>

Example

add iptunnel lb-dsr-tunnel-1 10.102.40.123 255.255.255.255 10.56.223.81/*

show iptunnel lb-dsr-tunnel-1

Parameters for configuring virtual servers

Name

A name for your new IP tunnel. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

Remote IP

The IP of the service that corresponds to the load-balanced server that you are configuring.

Remote Subnet Mask

The netmask of the subnet in which the remote IP is located.

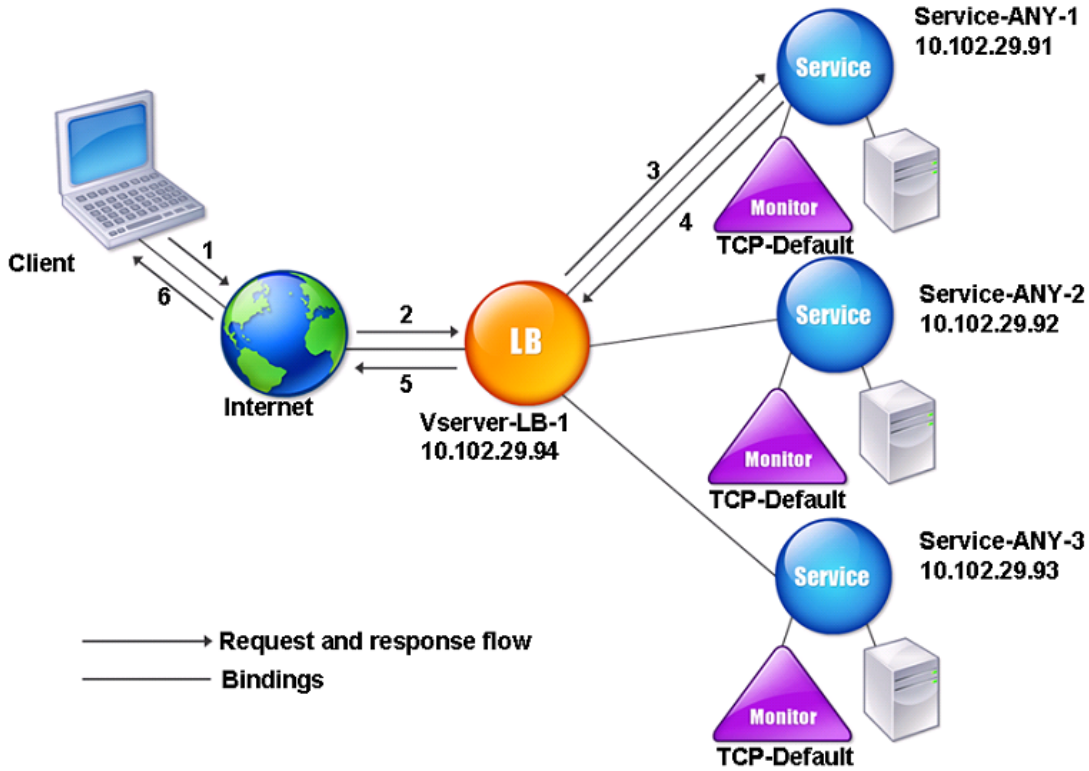
Local IP

The VIP of the load balancing virtual server.

Configuring Load Balancing in One-arm Mode

In a one-arm setup, you connect the NetScaler appliance to the network through a single interface. This is one of the simplest deployment scenarios, where the router, the servers and the appliance are all connected to the same switch. The client can access the server directly, bypassing the appliance, if the client knows the IP address of the server. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service, as is shown in the following diagram.

Figure 1. Entity Model for Load Balancing in One-Arm Mode



In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service. The following table lists the names and values of the entities configured on the appliance in one-arm mode.

Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY

	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
Monitors	TCP	None	None

The following diagram shows the load balancing entities and values of the parameters that need to be configured on the appliance.

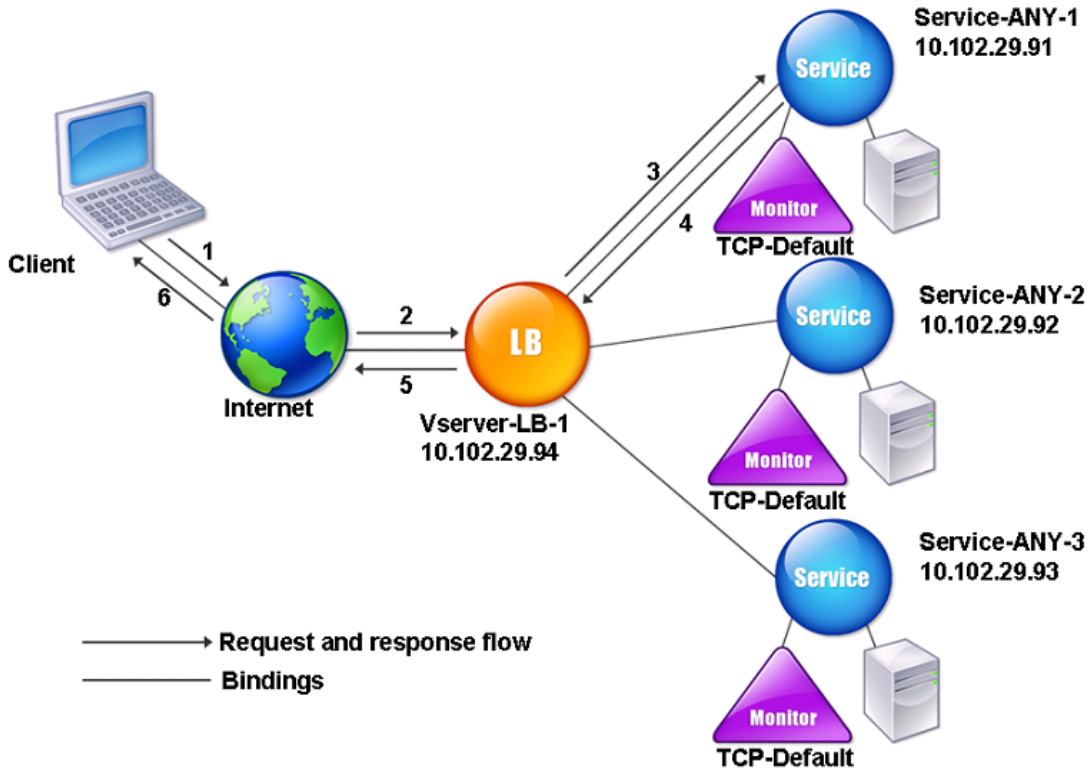


Figure 2. Entity Model for Load Balancing in One-Arm Mode

To configure a load balancing setup in one-arm mode, see Setting Up Basic Load Balancing.

Configuring Load Balancing in the Inline Mode

In an inline mode (also called two-arm mode) setup, you deploy the NetScaler appliance to the network through more than one interface. In the two-arm setup, the appliance is connected between the servers and the client. Traffic from clients passes through the appliance to access the load balanced server. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service. This is shown in the following diagram.

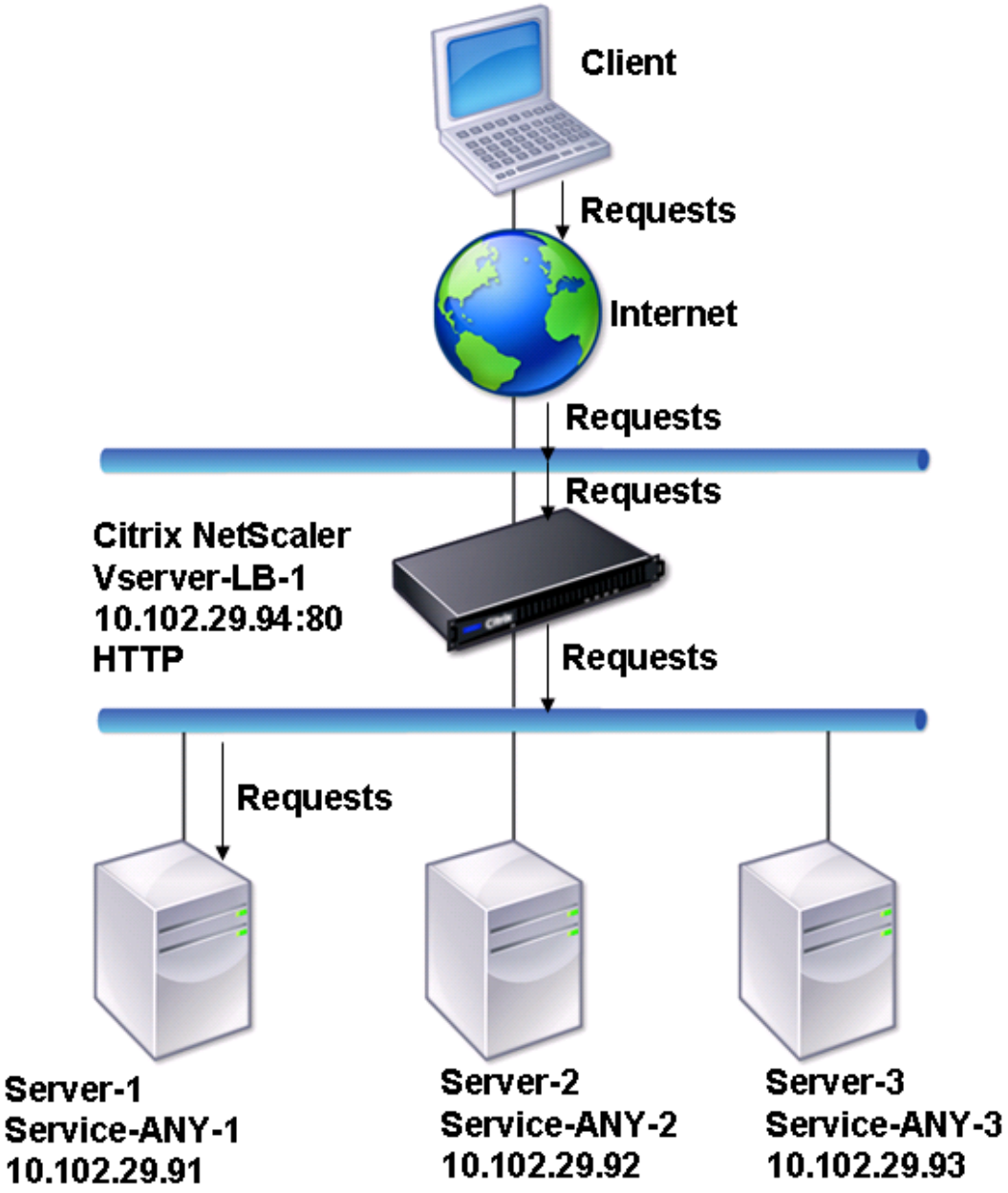


Figure 1. Load Balancing in Inline Mode

The configuration and the entity diagram for inline mode are the same as described in Configuring Load Balancing in One-arm Mode.

Load Balancing of Intrusion Detection System Servers

To enable the NetScaler appliance to support load balancing of intrusion detection system (IDS) servers, the IDS servers and clients must be connected through a switch that has port mirroring enabled. The client sends a request to the server. Because port mirroring is enabled on the switch, the request packets are copied or sent to the NetScaler appliance virtual server port. The appliance then uses the configured load balancing method to select an IDS server, as shown in the following diagram.

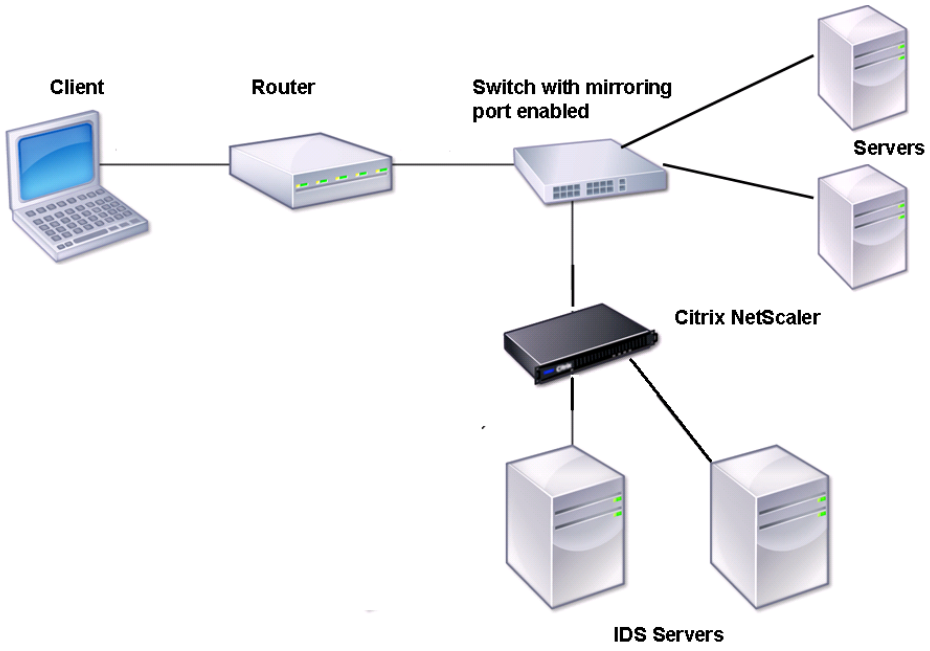


Figure 1. Topology of Load Balanced IDS Servers

Note: Currently, the appliance supports load balancing of passive IDS devices only.

As illustrated in the preceding diagram, the IDS load balancing setup functions as follows:

1. The client request is sent to the IDS server, and a switch with a mirroring port enabled forwards these packets to the IDS server. The source IP address is the IP address of the client, and the destination IP address is the IP address of the server. The source MAC address is the MAC address of the router, and the destination MAC address is the MAC address of the server.

2. The traffic that flows through the switch is mirrored to the appliance. The appliance uses the layer 3 information (source IP address and destination IP address) to forward

the packet to the selected IDS server without changing the source IP address or destination IP address. It modifies the source MAC address and the destination MAC address to the MAC address of the selected IDS server.

Note: When load balancing IDS servers, you can configure the SRCIPHASH, DESTIPHASH, or SRCIPDESTIPHASH load balancing methods. The SRCIPDESTIPHASH method is recommended because packets flowing from the client to a service on the appliance must be sent to a single IDS server.

Suppose Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to Vserver-LB-1. The virtual server balances the load on the services. The following table lists the names and values of the entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual server	Vserver-LB-1	*	*	ANY
Services	Service-ANY-1	10.102.29.101	*	ANY
	Service-ANY-2	10.102.29.102	*	ANY
	Service-ANY-3	10.102.29.103	*	ANY
Monitors	Ping	None	None	None

Note: You can use inline mode or one-arm mode for an IDS load balancing setup.

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

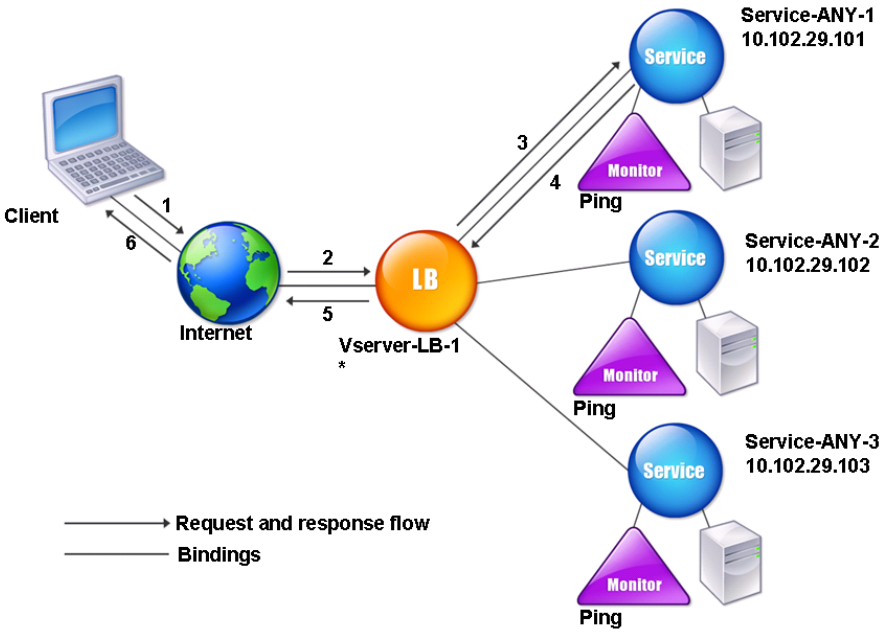


Figure 2. Entity Model for Load Balancing IDS Servers

To configure an IDS load balancing setup, you must first enable MAC-based forwarding. You must also disable layer 2 and layer 3 modes on the appliance.

To enable MAC-based forwarding by using the NetScaler command line

At the NetScaler command prompt, type:

enable ns mode <ConfigureMode>

Example

enable ns mode MAC

To enable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. On the **Settings** landing page, under **Modes and Features**, click **modes**.
3. In the **Configure Modes** dialog box, select the **MAC Based Forwarding** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, and then click **Yes**.

Next, see Setting Up Basic Load Balancing, to configure a basic load balancing setup.

After you configure the basic load balancing setup, you must customize it for IDS by configuring a supported load balancing method (such as the SRCIPDESTIP Hash method on a sessionless virtual server) and enabling MAC mode. The appliance does not maintain the state of the connection and only forwards the packets to the IDS servers without processing them. The destination IP address and port remains unchanged because the virtual server is in the MAC mode.

To configure LB method and redirection mode for a sessionless virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <vServerName> -**lbMethod** <LBMethodOption> -**m** <RedirectionMode> -**sessionless** <Value>

Example

set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -sessionless enabled

To configure LB method and redirection mode for a sessionless virtual server by using the configuration utility

1. In the left navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** pane, click the virtual server **Vserver-LB-1**, and then click **Open**.

3. On the **Method and Persistence** tab, under **LB Method**, select **Source IP Destination IP Hash**.
4. On the **Advanced** tab, under **Redirection Mode**, click **MAC Based**.
5. Select the **Sessionless** check box, and then click **OK**.

To set a service to use source IP address by using the NetScaler command line

At the NetScaler command prompt, type:

set service <ServiceName> -**usip** <Value>

Example

set service Service-ANY-1 -usip yes

To set a service to use source IP address by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. On the **Services** pane, select the service, **Service-ANY-1**, and then click **Open**.
3. On the **Advanced** tab, under **Settings**, select the **Use Source IP** check box.
4. Click **OK**.
5. Repeat steps 1-5 for the services **Service-ANY-2** and **Service-ANY-3**.

For USIP to function correctly, you must set it globally. For more information about configuring USIP globally, see the “IP Addressing” chapter in the *Citrix NetScaler Networking Guide* at http://support.citrix.com/article/CTX128671..

Isolating the Network Paths by Using Traffic Domains

A very common security requirement in a data center is to maintain network path isolation between the traffic of various applications or tenants. One application or tenant's traffic must be isolated from the traffic of other applications or tenants. For example, a financial services company would want to keep the traffic of its insurance department's applications separate from that of its financial services applications. In the past, this was easily achieved through physical separation of network service devices such as firewalls, load balancers, and IDP, and network monitoring and logical separation in the switching fabric.

As data center architectures evolve toward multi-tenant virtualized data centers, networking services in the aggregation layer of a data center are getting consolidated. This development has made network path isolation a critical component for network service devices and is driving the requirement for ADCs to be able to isolate traffic at the L4 to L7 levels. Furthermore, all the traffic of a particular tenant must go through a firewall before reaching the service layer.

To address the requirement of isolating the network paths, a NetScaler appliance identifies network domains and controls the traffic across the domains. The NetScaler solution has two main components: listen policies and shadow virtual servers.

Each network path to be isolated is assigned a virtual server on which a listen policy is defined so that the virtual server listens to traffic only from a specified traffic domain.

To isolate the traffic, listen policies can be based on a number of client parameters or their combinations, and the policies can be assigned priorities. The following table lists the parameters that can be used in listen policies for identifying the traffic.

Table 1. Client Parameters Used to Define Listen Policies

Category	Parameters
Ethernet protocol	Source MAC address, destination MAC address
Network interface	Network ID, receiving throughput, sending throughput, transmission throughput
IP protocol	Source IP address, destination IP address
IPv6 protocol	Source IPv6 address, destination IPv6 address
TCP protocol	Source port, destination port, maximum segment size, payload, and other options
UDP protocol	Source port, destination port
VLAN	ID

On the NetScaler appliance, a virtual server is configured for each domain, with a listen policy specifying that the virtual server is to listen only to traffic for that domain. Also configured for each domain is a shadow load balancing virtual server, which listens to traffic destined for any domain. Each of the shadow load balancing virtual servers has a wildcard (*) IP address and port, and its service type is set to ANY.

In each domain, a firewall for the domain is bound as a service to the shadow load balancing virtual server, which forwards all traffic through the firewall. Local traffic is forwarded to its destination, and traffic destined for another domain is forwarded to the firewall for that domain. The shadow load balancing virtual servers are configured for MAC mode redirection.

How Network Paths Are Isolated by Using NetScaler Traffic Domains

The following figure shows a typical traffic flow across domains. Consider the traffic flow within Network Domain 1, and between Network Domain 1 and Network Domain 2.

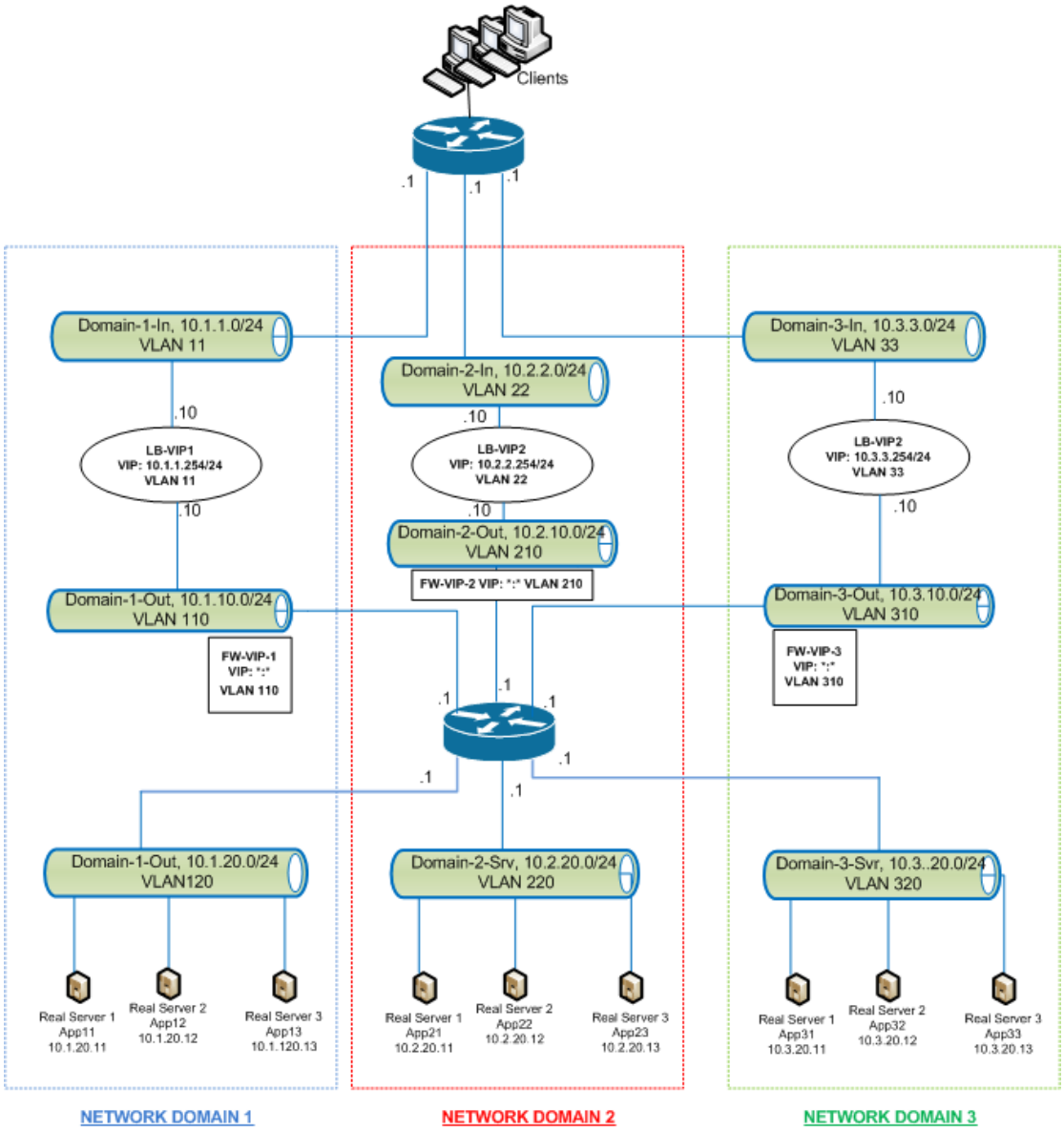


Figure 1. Network Path Isolation Using Traffic Domains

Traffic within Network Domain 1

Network Domain 1 has three VLANs: VLAN 11, VLAN110, and VLAN120. The following steps describe the traffic flow.

- A client from VLAN 11 sends a request for a service available from the service pool in VLAN 120.
- The load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110. The virtual server in VLAN 110 forwards the request to shadow load balancing virtual server FW-VIP-1.
- FW-VIP-1, which is configured to listen to traffic from VLAN 110, receives the request and forwards it to VLAN 120.

- The load balancing virtual server in VLAN 120 load balances the request to one of the physical servers, App11, App12, or App13.
- The response sent by the physical server returns by the same path to the client in VLAN 11.

This configuration ensures that traffic is always segregated inside the NetScaler for all the traffic that originates from a client.

Traffic between Network Domain 1 and Network Domain 2

Network Domain 1 has three VLANs: VLAN 11, VLAN 110, and VLAN 120. Network Domain 2 also has three VLANS: VLAN 22, VLAN 210, and VLAN 220. The following steps describe the traffic flow from VALN 11 to VLAN 22.

- A client from VLAN 11, which belongs to Network Domain 1, sends a request for a service available from the service pool in VLAN 220, which belongs to the Network Domain 2.
- In Network Domain 1, the load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110.
- Shadow load balancing virtual server FW-VIP-1, which is configured to listen to VLAN 110 traffic destined to any other domain, receives the request and forwards it to firewall virtual server FW-VIP-2 because the request is destined to a physical server in Network Domain 2.
- In Network Domain 2, FW-VIP-2 forwards the request to VLAN 220.
- The load balancing virtual server in VLAN 220 load balances the request to one of the physical servers, App21, App22, or App23.
- The response sent by the physical server returns by the same path through the firewall in Network Domain 2 and then to Network Domain 1 to reach the client in VLAN 11.

Configuring Traffic Domains

To configure network path isolation by using listen policies, do the following:

- Add listen policy expressions. Each expression specifies a domain to which traffic is destined. You can use the VLAN ID or other parameters to identify the traffic. For more details, see Client Parameters Used to Define Listen Policies.
- For each network domain, configure two virtual servers as follows:
  - Create a load balancing virtual server for which you specify a listen policy that identifies the traffic destined for this domain. You can specify the name of an expression created earlier, or you can create a new expression while creating the

virtual server.

- Create another load balancing virtual server, referred to as shadow virtual server, for which you specify a listen policy expression that applies to traffic destined for any domain. On this virtual server, set the service type to ANY and the IP address and port to an asterisk (*). Enable MAC-based forwarding on this virtual server.

- Enable the L2 Connection option on both the virtual servers.

  Generally, to identify a connection, the NetScaler uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.

- Add services representing the server pools in the domain, and bind them to the virtual server.

- Configure the firewall for each domain as a service, and bind all of the firewall services to the shadow virtual server.

To configure traffic domains by using the command line

At the NetScaler command prompt, type the following commands:

- add policy expression <expressionName> <listenPolicyExpression>

- add lb vserver <name> <serviceType> <ip> <port> **-l2conn** ON **-listenPolicy** <expressionName>

  Add a load balancing virtual server for each domain. This virtual server is for traffic of the same domain.

- add lb vserver <name> ANY * * **-l2conn** ON **-m** MAC **-listenPolicy** <expressionName>

  Add a shadow load balancing virtual server for each domain. This virtual server is for traffic of other domains.

Example

```
add policy expression e110 client.vlan.id==110
add policy expression e210 client.vlan.id==210
add policy expression e310 client.vlan.id==310
add policy expression e11 client.vlan.id==11
add policy expression e22 client.vlan.id==22
add policy expression e33 client.vlan.id==33

add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -listenPolicy e11 -cltTimeout 180 -l2Conn

add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE - listenPolicy e22 -cltTimeout 180 -l2Conn
```

```
add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE - listenPolicy e33 -cltTimeout 180 -l2Conn

add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e110 -Listenpri
add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e210 -Listenpri
add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e310 -Listenpri

add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport NO
add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport NO
add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport NO

bind lb vserver FW-VIP-1 RD-1
bind lb vserver FW-VIP-2 RD-2
bind lb vserver FW-VIP-3 RD-3
```

Parameters for configuring a service

name

Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

serverName

Either the name of a previously created server object, or the IP address of the load-balanced server that hosts this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually, and then select the server name instead of an IP address from the drop-down menu that is associated with this field.

If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

serviceType

The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP

port

Port on which the service listens. The port number must be a positive number not greater than 65535.

Parameters for configuring a virtual server

name

Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( )

IPAddress

IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.

serviceType

The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, RDP, and RTSP. Default: HTTP

port

Port on which the virtual server listens for client connections. The port number must be between 0-65535.

l2conn

The tuple used to identify a connection includes the layer 2 parameters

To configure traffic domains by using the configuration utility

1. Add services representing the servers, as described in Creating a Service.
2. Add each firewall as a service:
    a. In the navigation pane, expand **Load Balancing**, and then click **Services**.
    b. In the details pane, click **Add**.
    c. In the **Create Service** dialog box, specify values for the following parameters:
        - **Service Name***—The name that you assign to the service.
        - **Protocol***—Select ANY from the drop-down list.
        - **Server***—The firewall's IP address.
        - **Port***—Specify a value of 80.

        *A required parameter
    d. Click **Create**.
    e. From the **Services** pane, open the services you created and verify the settings.
3. Configure a load balancing virtual server.
    a. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
    b. In the details pane, click **Add**.
    c. In the **Create Virtual Server (Load Balancing)** dialog box, specify values for the following parameters, which are described in Creating a Virtual Server.
        - **Name***
        - **Protocol***
        - **IP Address***
        - **Port***

        *A required parameter
    d. On the **Services** tab, select the corresponding services.
    e. On the **Advanced** tab, select the **L2 Connection** check box and, for **Redirection Mode**, select **MAC Based**. Then, click the **Listen Policy** link and create the listen policy for the virtual server.
    f. Click **Create**.
4. Configure the shadow load balancing virtual server.
    a. For the shadow virtual server, specify

- **Protocol**—ANY
- **IP Address***—*
- **Port***—*

*A required parameter

b. Bind the firewall services to the shadow virtual server.

5. For each network domain, repeat steps 3 and 4.
6. From the **Load Balancing Virtual Servers** pane, open the virtual servers that you created and verify the settings.

Configuring XenDesktop for Load Balancing

For an improved performance in the delivery of virtual desktop applications, you can integrate the NetScaler appliance with Citrix® XenDesktop™ and use the NetScaler load balancing feature to distribute the load across the Web Interface servers and the Desktop Delivery Controller (DDC) servers.

Generally, you use XenDesktop in situations where applications are not compatible with running on a terminal server or XenApp, or if each virtual desktop has unique requirements. In such cases, you need one desktop host for each user that connects. However, the hosts can be pooled so that you need only one host for each currently connected user.

The core application service deployed for XenDesktop is the Desktop Delivery Controller (DDC). The DDC is installed on a server, and its main function is to register desktop hosts and broker client connections to them.

The DDC also authenticates users and manages the assembly of the users' virtual desktop environments by controlling the state of the desktops, and starting and stopping the desktops.

Generally, multiple DDCs are installed to enhance availability.

The Web Interface servers provide secure access to virtual desktops. The Web Interface is the initial connection portal to the Desktop Delivery Controller (DDC). The Web browser on the user's device sends information to the Web server, which communicates with the server farm to provide the user with access to the virtual desktop.

The following figure shows the topology of NetScaler working with XenDesktop.

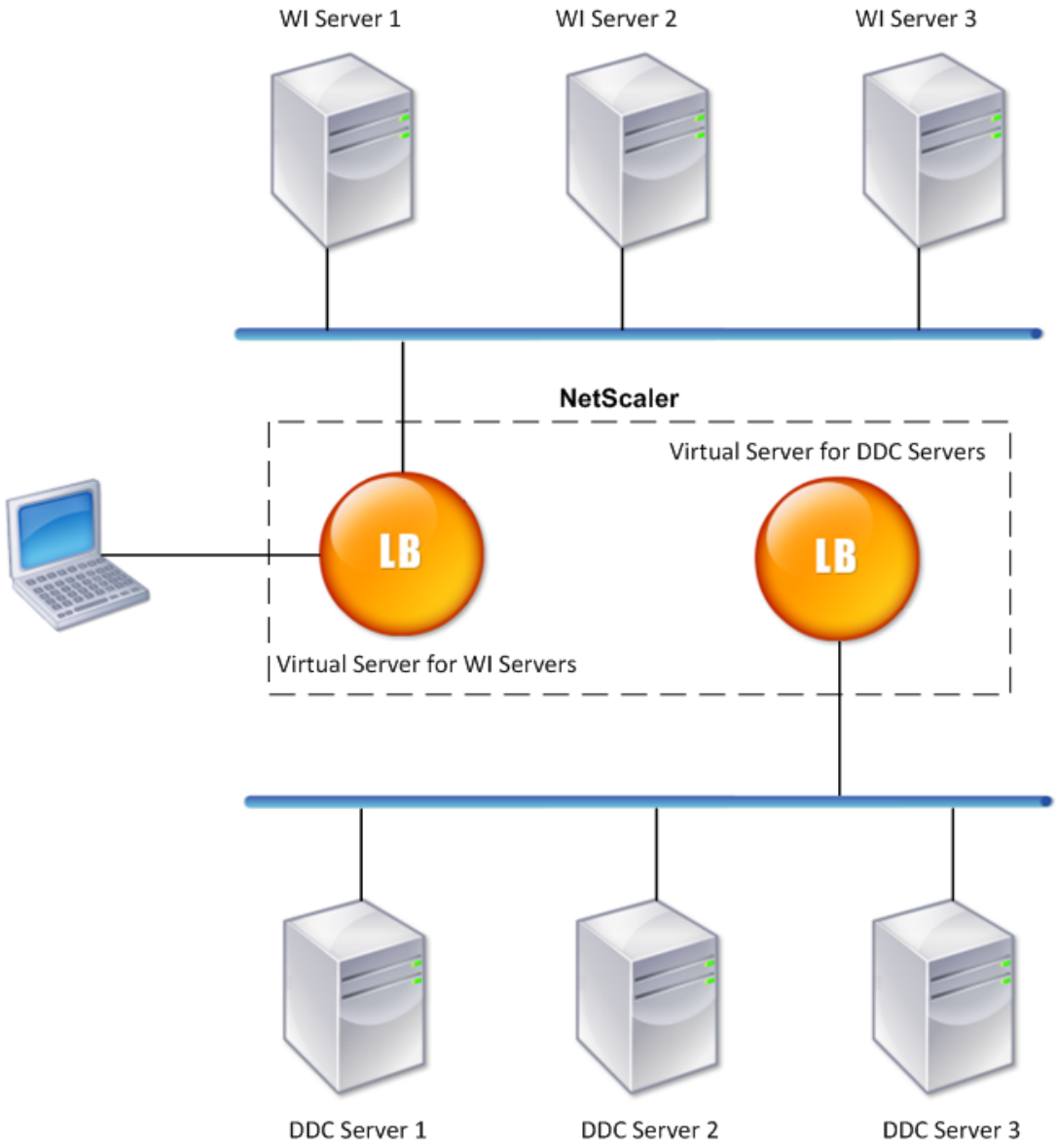


Figure 1. **Load Balancing of XenDesktop**

Note: Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication between the NetScaler and the DDC servers even though you use the SSL protocol for communication with the client.

A wizard is available for configuring basic load balancing in a XenDesktop deployment. You can use the wizard to configure Web interface servers and a virtual server for them, and DDC servers and a virtual server for them. The virtual servers that you configure are bound to services specified as Web Interface services and DDC services. Each virtual server is configured with the default load balancing method, and the default features are enabled. A monitor is created and bound to each virtual server.

The wizard creates a basic setup, with default values for options such as the load balancing method, policies, persistence, and advanced settings. You can change any of the values if necessary.

To configure load balancing for XenDesktop by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the **Getting Started** group, click **Load balancing wizard for Citrix XenDesktop**.
3. Follow the instructions presented by the wizard.

Configuring XenApp for Load Balancing

For efficient delivery of applications, you can integrate the NetScaler appliance with Citrix® XenApp™ and use the NetScaler load balancing feature to distribute the load across the XenApp server farms. The following figure is a topology diagram of such a setup.

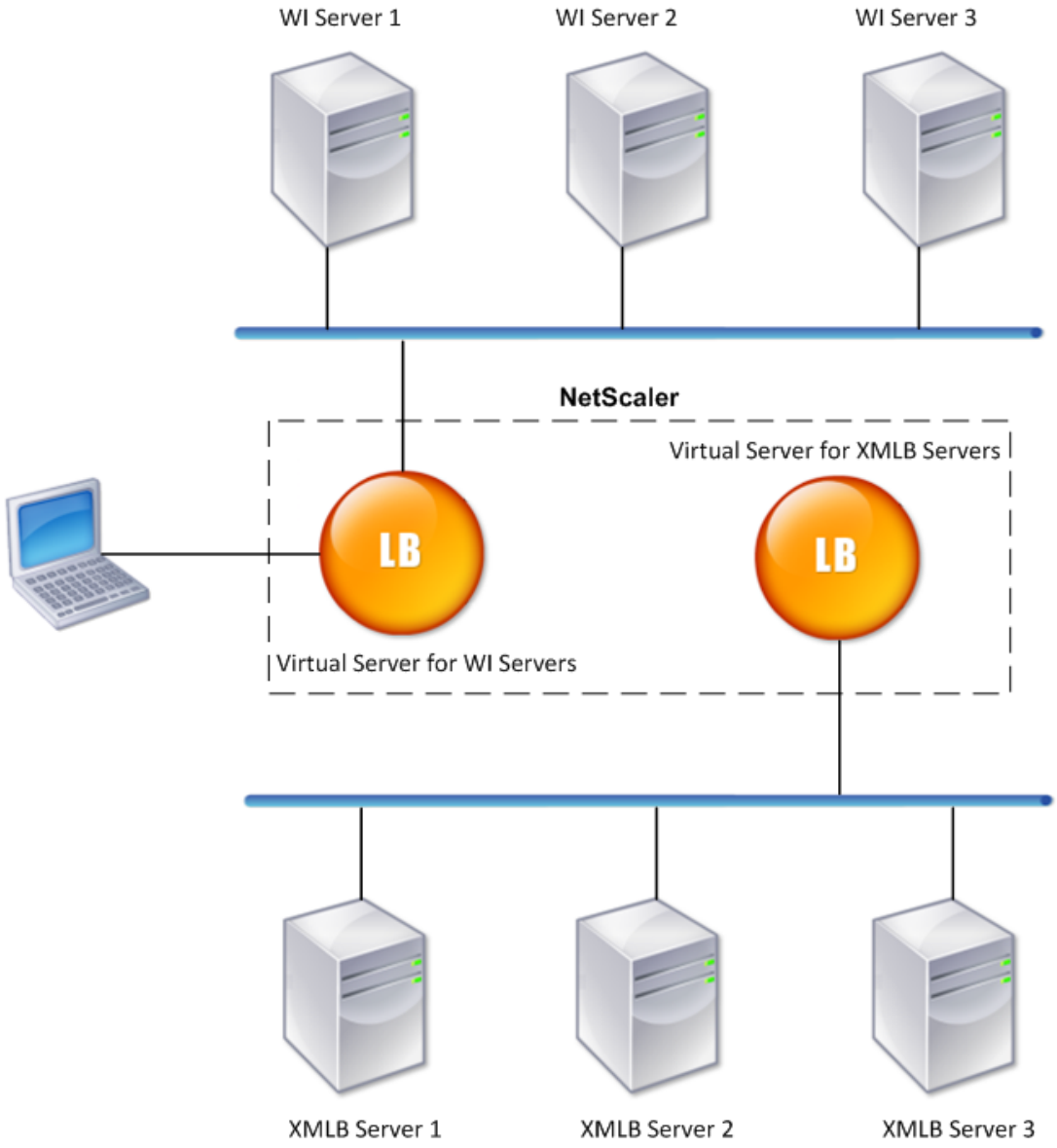


Figure 1. Load Balancing of XenApp

The Web Interface servers provide secure access to XenApp application resources through the user's Web browser. The Web Interface client presents to the users all the resources, such as applications, content, and desktops that are made available in the XenApp server farms. Users can access the published resources through a standard Web browser or through the Citrix online plug-in.

The Web browser on the user's device sends information to the Web server, which communicates with the servers on the server farm to provide the user with access to the resources.

The Web Interface and the XML Broker are complementary services. The Web Interface provides users with access to applications, and the XML Broker evaluates the user's permissions to determine which applications appear in the Web Interface.

The XML service is installed on all the servers in the server farm. The XML service specified in the Web Interface functions as an XML broker. On the basis of the user credentials passed by the Web Interface server, the XML Broker server sends a list of applications accessible to the user.

In large enterprises where multiple Web Interface servers and XML Broker servers are deployed, Citrix recommends load balancing these servers by using NetScaler. Configure one virtual server to load balance all of the Web Interface servers and another for all of the XML Broker servers. The load balancing method and other features can be configured on the virtual server as required.

Note: Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication between the NetScaler and the WI servers even though you use the SSL protocol for communication with the client.

The configuration utility provides a wizard for setting up basic load balancing for XenApp.

Through this wizard, you can configure Web Interface servers and a virtual server for them, and XML Broker servers and a virtual server for them. You can also specify the site through which the status of Web Interface servers can be monitored and the software application used to monitor the status of the XML Broker servers.

When you complete the wizard, a basic load balancing setup is configured on the NetScaler. The specified virtual servers are created and bound to the services specified as Web Interface services and XML Broker services. Each virtual server is configured with the default load balancing method, and the default features are enabled. A monitor is created and bound to each virtual server.

The wizard creates a basic setup with default values for options such as the load balancing method, policies, persistence, and advanced settings. You can change any of the values if necessary.

To configure load balancing for XenApp by using the configuration utility

1. In the navigation pane, click **Load Balancing**.
2. In the **Getting Started** group, click **Load balancing wizard for Citrix XenApp**.
3. Follow the instructions presented by the wizard.

Troubleshooting Common Problems

Below are a few tips for troubleshooting common problems when configuring load balancing on the NetScaler appliance.

- When a metric bound to a monitor is present in the local and custom metric tables, add the local prefix to the metric name if the metric is chosen from the local metric table. If the metric is chosen from the custom table, no prefix needs to be added.
- If the metric table is modified (for example, if the OID for the metric is changed), the change is reflected in the monitoring table. SNMP queries originating from the monitor then use the new OID.
- Load monitors cannot decide the state of the service. Therefore, setting a weight on the load monitors is inappropriate.
- If multiple load monitors are bound to a service, then the load on the service is the sum of all the values on the load monitors bound to it. For load balancing to work properly, you must bind the same set of monitors to all the services.
- When you bind a service to a virtual server where the LB method is CUSTOMLOAD, and if the service is up, then the virtual server is put to initial round robin. It continues to be in round robin if the service has no custom load monitors, or if at least one of the custom load monitors is not up.
- If you disable a load monitor bound to the service, and if the service is bound to a virtual server, then the virtual server goes to round robin.
- If you disable a metric-based binding, and if this is the last active metric, then the specific virtual server goes to round robin. A metric is disabled by setting the metric threshold to zero.
- When a metric bound to a monitor crosses the threshold value, then that particular service is not considered for load balancing.
- If all the services have reached the threshold, then the virtual server goes into round robin and an error message “5xx - server busy error” is received.

    - All the services that are bound to a virtual server where the load balancing method is CUSTOMLOAD must have load monitors bound to them.
    - The OIDs must be scalar variables.
    - For successful load balancing, the interval must be as low as possible. If the interval is high, the time period for retrieving the load value increases. As a result, load balancing takes place using improper values.
    - The CUSTOMLOAD load balancing method also follows startup round robin.

- A user cannot modify the local table.
- A maximum of 10 metrics from a custom table can be bound to the monitor.

NetScaler Web 2.0 Push

Modern web applications, also referred to as web 2.0 applications, provide highly responsive interfaces that generate asynchronous updates that can impose an additional load on a server. Typically, asynchronous notifications are sent by using HTTP and server push techniques, such as long-polling and streaming response, which enable servers to push the notifications to clients. These techniques require the servers to maintain a large number of TCP/IP connections, which provides low latency but results in low bandwidth. As the number of clients increases, the servers are overloaded with connections kept open for each client. Further, the large number of connections terminating on the server requires kernel resources and memory for data structures like protocol control blocks, socket descriptors, and socket buffers.

With the NetScaler Web 2.0 push feature, you can use the NetScaler appliance as a proxy server to offload long-lived client TCP connections and maintain relatively fewer, reusable connections to the server. NetScaler Web 2.0 push is application agnostic, with the flexibility to work seamlessly with various technologies and configurations used for asynchronous messaging. It can be extended to co-exist with developing technologies, and it preserves backward compatibility. NetScaler Web 2.0 Push is also scalable, with support for multiple NetScaler appliances.

With the NetScaler Web 2.0 push feature, the NetScaler appliance multiplexes and manages the exchange of data reliably and securely, reducing the number of server-side connections across potentially millions of persistent client connections. For every HTTP, HTTPS, or SSL transaction, the appliance can de-link and rebalance the server farm to distribute client requests across multiple servers.

The NetScaler Web 2.0 Push feature reduces the number of server-side connections across millions of persistent client connections.

Web 2.0 Push Applications

With modern Web applications, termed broadly as Web 2.0 applications, servers use AJAX technologies such as polling to maintain up-to-date information about the client. Polling enables an AJAX application to periodically poll the server for updates. For example, a chat based application can poll a Web server every 10 seconds for any chat updates. To get such updates from the Web server, the client browser periodically opens a connection to the Web server.

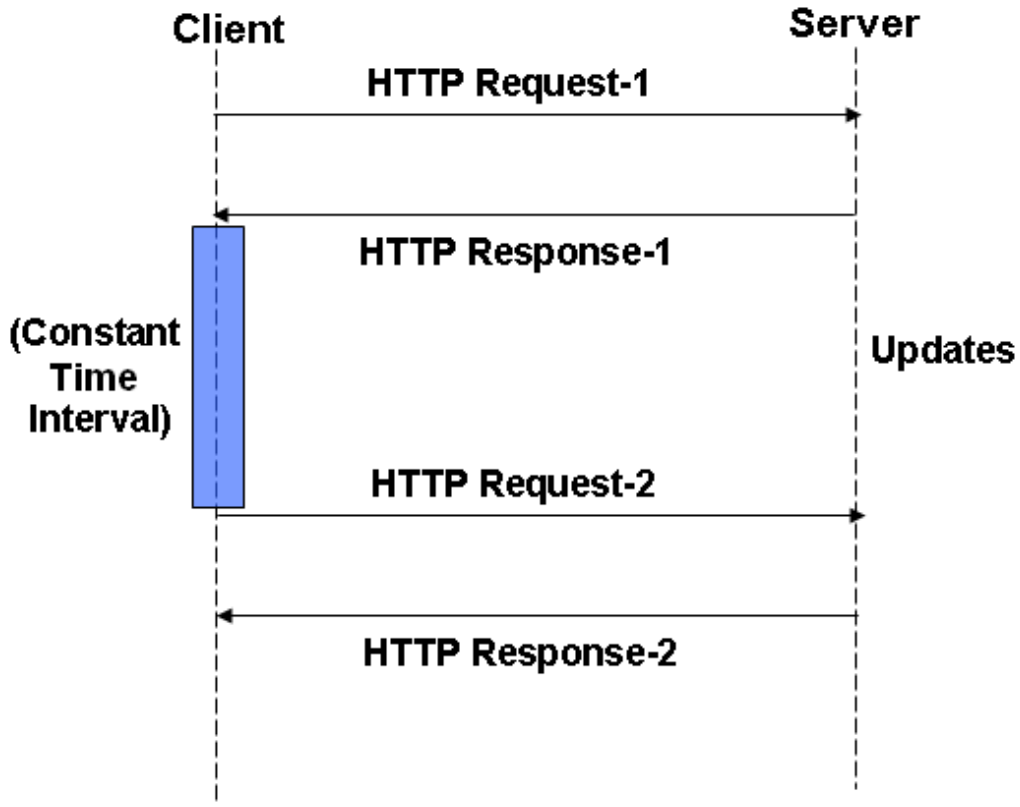


Figure 1. Polling Technique

Such frequent polling can overload the server. Also, if you deploy the AJAX application on a Web server with low resources and a large number of simultaneous users poll the server for updates, the network can become saturated, with significant degradation in the server performance. And if there is no update from the server, the client requests overload the server for a void response.

To avoid such problems, server push technology often uses a long polling technique. Long polling enables the client application to open a persistent connection to the server and wait for the server to push updates when available.

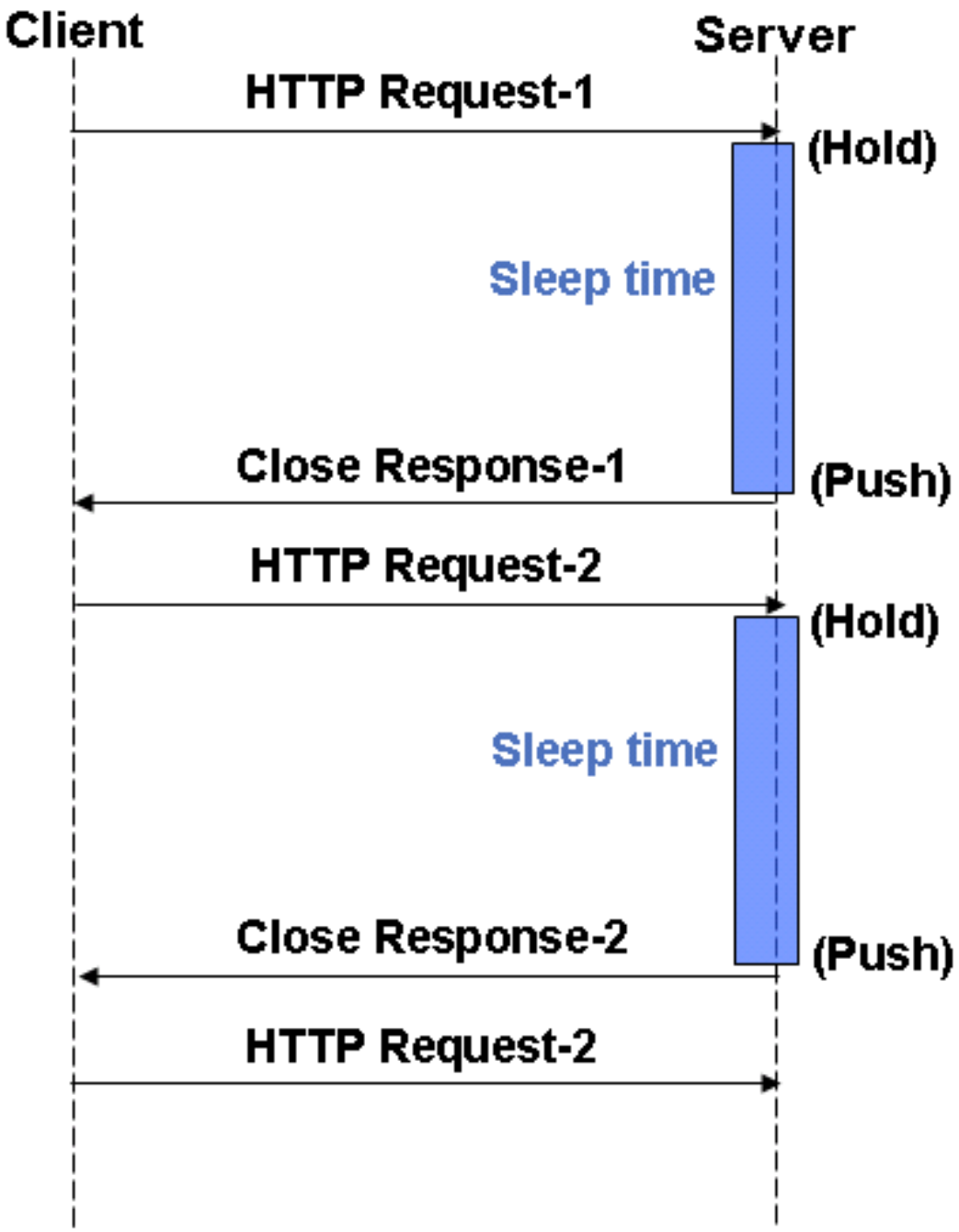


Figure 2. Long Polling Technique

If your server supports asynchronous request processing, long polling is a scalable technique. However, long polling can hold the server connections until updates are available. For example, if 1,000 AJAX applications open one long polled connection, 1,000 threads hold the server while waiting for updates.

Another technique, called HTTP streaming, is identical to the long polling technique except that the connection is not closed after the server pushes the updates. The AJAX application sends a single request and receives chunks of responses (partial responses) over the same connection. With HTTP streaming, the browsers and server do not open or close the connection. Therefore, HTTP streaming significantly reduces the network latency.

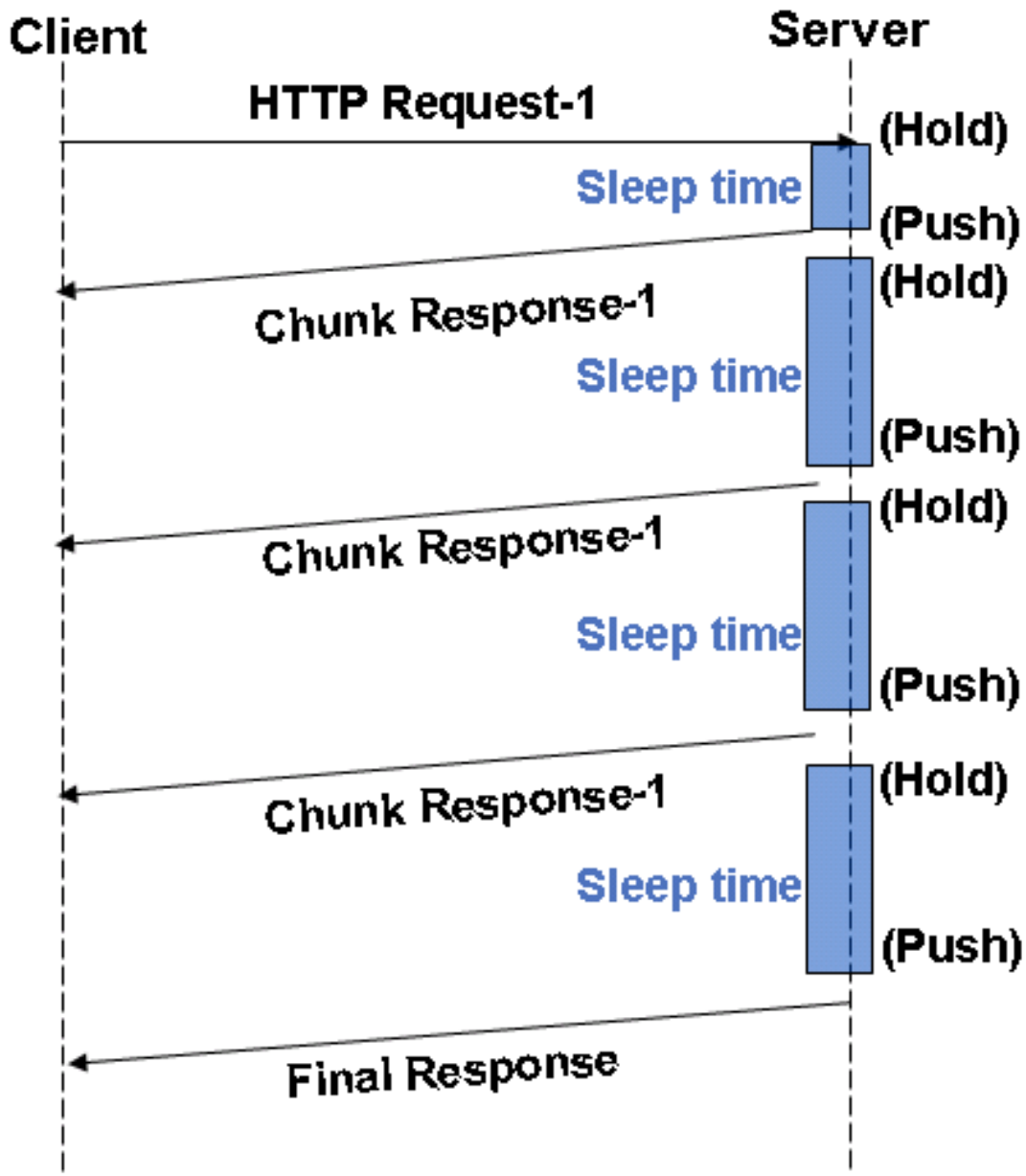


Figure 3. HTTP Streaming Technique

In HTTP streaming, as in long polling techniques, if the server frequently pushes updates, the performance of the network and the AJAX applications are significantly degraded and the client may lose the updates. If your AJAX application opens both long polling and HTTP streaming connections to the same Web server, other AJAX applications cannot open connections to the server, because the browser blocks such connections.

NetScaler Web 2.0 push uses connection labeling to overcome the limitations of long polling and HTTP streaming.

How Web 2.0 Push Works

The NetScaler Web 2.0 push feature enables the server to label a client connection and subsequently identify and send data over that labeled connection. With NetScaler Web 2.0 push enabled, the client first establishes a TCP/IP connection and connects to the NetScaler appliance. The appliance uses the configured load balancing method or content switching policy to select a Web server to which to open a connection and send the client request. The server interacts with the client and uses either authentication or a previously established cookie to identify the client.

When the NetScaler appliance receives a request with push enabled, it initiates the labeling protocol with the Web server. This protocol enables the Web server to label the connection and defer the response. The protocol also enables the server to process other requests without invoking push processing. The Web server (referred to as a *notification server*) uses the label to send updates to the client through the NetScaler appliance when the updates become available. Servers can choose to push multiple updates over a single TCP connection or open one connection per update.

Note: The set of Web servers that respond to requests from the NetScaler does not necessarily include the notification servers that push updates to client.

A central component of a NetScaler Web 2.0 push configuration is a push virtual server, which is a load balancing virtual server with service type PUSH or SSL_PUSH. The NetScaler appliance uses the push virtual server to expose the message push protocol to the Web servers. A server uses the protocol to push asynchronous messages to connected clients. A push virtual server exposes a simple REST interface for posting updates.

Important: For the NetScaler Web 2.0 Push feature to work correctly, you must configure the NetScaler appliance as a proxy for the traffic between the clients and servers. You can use multiple NetScaler appliances to scale up your connection management.

For each transaction, the NetScaler Web 2.0 push feature maintains a state machine, which manages the actions of the transaction. The state machine has the following states:

- Waiting for Request State (Q)-A connection has been established between the client and the NetScaler appliance. The appliance waits in this state until the client sends a request.
- Waiting for Server Response State (R) -A request has been received from the client and forwarded to a Web server. The appliance waits in this state for the server to respond.
- Waiting for Asynchronous Messages State (A) -The appliance is waiting for asynchronous messages that the notification servers push to the push virtual server.

Until the client establishes a connection with the NetScaler appliance's load balancing or content switching virtual server, the initial state of the transaction is Q. When the appliance receives a request, it forwards the request to the server, and the transaction moves to state R.

If the appliance receives a deferred response (also called a *labeled response*), the transaction moves to state A. In this state, if the appliance receives a push message

through the message push protocol, it processes the message and forwards the message to the client. If this message is marked as the last message, the appliance closes the transaction and moves to state Q. If not, the transaction remains in state A.

The push virtual server can manage long-polling and streaming responses from the server. Each update that the server sends to the push virtual server has a flag (with query parameters) that indicates whether there are updates from the server. When the flag indicates that the updates from the server are unavailable, the NetScaler appliance performs one of the following functions:

1. If the client uses HTTP 1.1 protocol and multiple updates are received from the server, the appliance sends a chunked response to the client and appends a zero chunk to the final response. If the first response itself has the flag set, the content length itself is sent as the response.

2. If the client uses HTTP 1.0 protocol and multiple updates are received from the server, then just the contents of the chunked response or the body of the content length response is sent to the client and the connection is terminated. If the first response itself has the flag set then, the content length itself is sent as the response.

The appliance sends a content-length response regardless of which HTTP version the client uses. The connection-labeling and message-push protocols, which identify the client and the server connections, provide the basic functionality of the NetScaler Web 2.0 push feature.

Understanding NetScaler Web 2.0 Push Protocol

For the NetScaler Web 2.0 Push feature to work correctly, the NetScaler appliance must label the client connection and then identify and send the deferred response from the server over the labeled connection. For this purpose, the Web 2.0 push feature uses the connection labeling and the message push protocols.

Connection Labeling Protocol

The connection labeling protocol is used between the server and the NetScaler appliance to label the client connection. After a label is negotiated, the Web server includes the label in the update that is sent to the client.

The appliance forwards a request to the server after adding an X-NS-PUSHVSERVER header containing the IP address and port of the push virtual server. The server either responds to this request with an HTTP response or defers the response. If the server defers the response, it labels the connection with an X-NS-DEFERRABLE header, which indicates that the connection is deferred.

A policy configured on the load balancing or content switching virtual server enables the NetScaler appliance to extract the label from the response. The appliance uses the information in the label to send the push message (update) to the push virtual server, which sends the response on the corresponding client connection.

Note: For any update from the Web server, the NetScaler does not support rewrite and compression.

When a server receives a request that it is deferrable, it sends an HTTP 200 OK response with the X-NS-DEFERRABLE header, which indicates to the NetScaler appliance that the push feature should be applied to the request. The appliance removes the X¤NS-DEFERRABLE header, sends the response to the client, and waits for updates. For example:

```
HTTP/1.1 200 OK
Date: Wed, 25 Aug 2010 18:22:47 GMT
Server: Apache/2.0.61 (FreeBSD) PHP/5.2.5 with Suhosin-Patch mod_ssl/2.0.61
OpenSSL/0.9.8e mod_perl/2.0.3 Perl/v5.8.8
X-NS-DEFERRABLE: YES
X-NS-SERVERLABEL: 04c2442bcb7c4b5f826d41a623e374e!
Content-Length: 0
Content-Type: text/plain;charset=UTF-8
```

Message Push Protocol

The message push protocol is used between a notification server and the NetScaler appliance to enable the notification server to send a notification to a previously labeled client connection.

Web servers use the message push protocol to push asynchronous messages to connected clients. The push protocol is built as a REST interface, exposed through the push virtual server on the NetScaler appliance. The server connects to the push virtual server and sends a request to the appliance. The BODY of the request contains the payload to be sent to the client. Additionally, the request identifies the label for the target client connection and the last message of the response.

When the NetScaler appliance receives the deferred response from the server, it sends the response to the client as a single HTTP chunk and sends a 200 OK response with the XML information to the server. If the message is marked as the last message of the response, the NetScaler also closes the HTTP response on the server.

Note: If the NetScaler is aware of the content length, it may send a response specifying the Content-Length, instead of a chunk. This enables the NetScaler to manage both HTTP streaming and long-polling responses.

Notification from Server to Push Server

```
POST /CLIENT/V10/04c2442bcb7c4b5f826d41a623e374e!?MSG_END=0  HTTP/1.1
Host: 10.102.80.66:8080
Content-Length: 6
```

Response from Push vserver to the server

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="UTF-8"
Content-Length: 130
<?xml version="1.0" encoding="UTF-8"?>
<CLIENTINFO>
  <CLIENT ID="04c2442bcb7c4b5f826d41a623e374e!" INFO="SUCCESS" />
</CLIENTINFO>
```

Configuring Web 2.0 Push

To configure NetScaler Web 2.0 push, you must first enable the feature. Then, create a push virtual server and associate it with a load balancing or content switching virtual server. Once you have a working configuration, you can customize it to suit your deployment.

You can also monitor the Web 2.0 push configuration by viewing statistics about the push virtual server and the other entities, such as the load balancing or the content switching virtual servers, that are part of the configuration.

Enabling NetScaler Web 2.0 Push

You have to enable the NetScaler Web 2.0 push feature before you can use it. Before enabling the feature, you must have the appropriate license installed on the NetScaler appliance. With the feature disabled, you can configure NetScaler Web 2.0 push entities, such as the push virtual server, but the entities will not work until the feature is enabled.

To enable NetScaler Web 2.0 push by using the NetScaler command line

At the NetScaler command prompt, type:

enable feature push

If NetScaler Web 2.0 Push is not licensed or disabled, the push virtual server state is DOWN.

To enable NetScaler Web 2.0 Push by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change advanced features**.
3. In the **Configure advanced features** dialog box, select the **NetScaler Push** check box, and then click **OK**.
4. At the **Enable/Disable Feature(s)?** prompt, click **Yes**.

Creating a NetScaler Web 2.0 Push Virtual Server

A push virtual server enables the NetScaler appliance to multiplex and manage the exchange of data (server push) reliably, securely, and in a scalable manner. It enables the notification server to send a notification to a previously labeled client connection by using the message push protocol. The notification servers push the out-of-band updates to the push virtual server. When the clients access the load balancing or the content switching virtual servers, the push virtual server uses the labeling protocol to label the deferred clients.

You can add, modify, and remove push virtual servers, however, you cannot bind services to the push virtual server.

To create a NetScaler Web 2.0 Push virtual server by using the NetScaler command line

At a NetScaler command prompt, type the following commands to create a push virtual server and verify the configuration:

- add lb vserver <name> <serviceType> <IPAddress> <Port>
- show lb vserver <name>

Example

```
add lb vserver Vserver-Push-1 PUSH 10.102.29.162 80
show lb vserver Vserver-Push-1
```

Parameters for creating a push virtual server

name

The name of the push virtual server being created. This alphanumeric string is required and cannot be changed after the virtual server is created. Must not exceed 127 characters, and leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ).

IPAddress

The IP address of the push virtual server being configured. This is a mandatory parameter.

serviceType

The type of data transferred between the NetScaler appliance and the client. SSL_PUSH service type encrypts the traffic between the NetScaler and the client. Use PUSH service type for HTTP requests. Possible values: PUSH, SSL_PUSH.

Note: For an SSL_PUSH virtual server, you need to bind a certificate-key pair. For details about binding a certificate-key pair to the virtual server, see Secure Sockets Layer (SSL) Acceleration.

port

The TCP port on which the virtual server listens. This is a mandatory argument. Must be a positive number not greater than 65535. Minimum value: 1.

To create a NetScaler Web 2.0 Push virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Name**, **Port**, and **IP Address** text boxes, type a name for the push virtual server, a port, and an IP address (for example, **Vserver-Push-1**, **80**, and **10.102.29.162**).
4. In **Protocol**, select either **SSL_PUSH** or **PUSH**.
5. Click **Create**, and then click **Close**. The push virtual server you created appears in the **Load Balancing Virtual Servers** pane.

To remove a push virtual server, use the rm lb vserver command that takes only the **name** parameter.

Configuring a Load Balancing or Content Switching Virtual Server

After creating a push virtual server, you need to associate it with the load balancing or content switching virtual servers. For details about creating a load balancing virtual server, see Creating a Virtual Server. Also, for details about creating a content switching setup, see Creating Content Switching Virtual Servers.

Once you have created the load balancing or content switching virtual servers, you must associate them with the push virtual server.

To configure a load balancing virtual server or content switching virtual server for NetScaler Web 2.0 push by using the NetScaler command line

At a NetScaler command prompt, type the following commands to configure a load balancing virtual server or content switching virtual server for NetScaler Web 2.0 push and verify the configuration:

- (set lb vserver | set cs vserver) <name> <ServiceType> <IPAddress> <Port> **-push** (ENABLED | DISABLED) **-pushVserver** <PushVservername> **-pushLabel** <Expression> **-pushMultiClients** (YES | NO)
- (show lb vserver | show cs vserver) <name>

Examples

set lb vserver Vserver-LB-1 HTTP 10.102.29.161 80 -push ENABLED - pushVserver PushVserver1 -pushLabel "H

show lb vserver Vserver-LB-1

set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -push ENABLED - pushVserver PushVserver1 -pushLabel "H

show cs vserver Vserver-CS-1

To modify or remove a load balancing or content switching virtual server by using the NetScaler command line

- To modify a virtual server, type the set lb vserver or set cs vserver command, the name of the virtual server, and the parameters to be changed, with their new values.
- To remove a virtual server, type the rm lb vserver or rm cs vserver command and the name of the load balancing or content switching virtual server.

Parameters for configuring a load balancing virtual server

push

Enable the NetScaler appliance to process traffic with the push virtual server. Possible values: ENABLED, DISABLED. Default: DISABLED

pushVserver

The name of the load balancing virtual server, of type PUSH or SSL_PUSH, to which the server pushes updates received on the client-facing load balancing virtual server.

pushLabel

An expression for extracting the label from the response from server. The string can be either an existing rule name (configured by using the add rule command) or an in-line expression with a maximum of 64 characters. Default value: NONE.

pushMultiClients

Allow multiple Web 2.0 connections from the same client to connect to the virtual server and expect updates. Possible values: YES, NO. Default value: NO.

To create a load balancing virtual server for NetScaler Web 2.0 Push by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure push virtual server (for example, **Vserver-LB-1**), and click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, select the **Enable Push** check box and in the **Push Virtual Server** list, select the push virtual server (for example, **Vserver-Push-1**) and click **OK**.

   Note: To create a content switching virtual server for NetScaler Web 2.0 Push by using the configuration utility, in the navigation pane, expand **Content Switching**, click **Virtual Servers**, Then, perform steps 2 and 3.

Monitoring the Configuration

To monitor the NetScaler Web 2.0 push configuration, you need to view the statistics of the push virtual servers and load balancing entities. This is useful for troubleshooting.

For instructions on how to display statistics of load balancing entities, see Load Balancing. Available statistics include labeled connections, push labeled connections, and deferred requests.

To view the properties of the push virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

show lb vserver <PushVserverName>

Example

show lb vserver Vserver-Push-1

Customizing the NetScaler Web 2.0 Push Configuration

Once your basic Web 2.0 Push configuration is operational, you can customize it by setting a time-out value for idle client connections and configuring URL redirects.

Setting a Time-out Value for Idle Client Connections

Once a client connects to the push virtual server, you can configure the virtual server to close any idle client connections after a configured time period.

To configure a time-out value, use the **cltTimeout** parameter, which specifies the time, in seconds, after which the NetScaler appliance closes any idle client connections. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

To set a time out value for idle client connections by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <PushVserverName> [**-cltTimeout** <secs>]

Example

```
set lb vserver Vserver-Push-1 -cltTimeout 100
```

To set a time-out value for idle client connections by using the configuration utility

1. In the navigation pane, expand **Load Balancing** and click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure virtual server port insertion (for example, **Vserver-Push-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, click the **Advanced** tab.
4. In the **Client Time-out (secs)** text box, type the timeout value (for example, **100**).
5. Click **OK**.

Redirecting Client Requests to an Alternative URL

You can configure a URL to which to redirect HTTP or HTTPS client requests when the push virtual server is down or disabled. This URL can be a local or a remote link. The NetScaler appliance uses HTTP 302 redirect to redirect client requests.

Redirects can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only a domain name (relative URL), the incoming URL is appended to the domain configured in the redirect URL.

The domain specified in the redirect URL must not be the same as the domain specified in the domain name argument of a content switching policy. If the same domain is specified in both arguments, the request is redirected continuously to the same unavailable virtual server in the NetScaler appliance, and the user cannot get the requested content.

To configure a virtual server to redirect the client request to a URL by using the NetScaler command line

At the NetScaler command prompt, type:

set lb vserver <name> **-redirectURL** URLValue

Example

set lb vserver Vserver-Push-1 -redirectURL http://www.newdomain.com/mysite/maintenance

To configure a virtual server to redirect the client request to a URL by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the push virtual server for which you want to configure redirect URL (for example, **Vserver-Push-1**), and then click **Open**.
3. On the **Advanced** tab, in the **Redirect URL** text box, type the URL (for example, **http://www.newdomain.com/mysite/maintenance**).
4. Click **OK**.

Pattern Sets

A pattern set is an array of indexed patterns that you configure on the Citrix® NetScaler® appliance. Pattern sets are used for string matching during default syntax policy evaluation. The NetScaler appliance provides you with a set of default syntax expression operators that you can use to compare a string in a packet with the patterns that are indexed and stored in a pattern set.

You can configure the appliance to compare the string that is identified in a packet with one or more patterns in the pattern set by using a simple default syntax expression combined with an operator. Additionally, after you create a pattern set, you can use the pattern set in multiple default syntax policies. Therefore, pattern sets eliminate the need for you to configure compound default syntax expressions that perform string matching with multiple OR operations (one expression for each comparison). This also reduces the consumption of appliance resources in terms of memory and the number of expressions that the appliance has to evaluate.

First, you create a pattern set and bind patterns to it. Then, when you configure a policy for comparing a string in a packet with the pattern set, you use an appropriate operator and pass the name of the pattern set as an argument.

How String Matching with a Pattern Set Works

A pattern set contains a set of patterns, and each pattern is assigned a unique index. During policy evaluation, the operator compares the string that is identified in the packet with the patterns defined in the pattern set until a match is found. Then, depending on its function, the operator returns either a Boolean value that indicates whether or not a matching pattern was found or the index of the pattern that matches the string.

Following is an example pattern set, "imagetypes," that defines a set of image file extensions. The two example expressions that follow the definition of the pattern set demonstrate how the two types of operators work. The first example describes the functioning of an operator that returns a Boolean value. The second example describes the functioning of an operator that returns the index of the pattern that matches the string.

Table 1. Example Pattern Set "imagetypes"

Bound Pattern	Index
bmp	1
jpeg	2
png	3
gif	4
tiff	5
svg	6

Example 1. The following default syntax expression identifies the URL suffix in an HTTP packet. The operator EQUALS_ANY compares the suffix to the image file extensions defined in imagetypes and returns a Boolean value to indicate the result of the comparison.

```
HTTP.REQ.URL.SUFFIX.EQUALS_ANY("imagetypes")
```

If the URL in the incoming HTTP packet that the appliance is evaluating is http://www.example.com/homepageicon.jpeg, the operator returns a Boolean TRUE.

Example 2. The following default syntax expression identifies the URL suffix in an HTTP packet. The operator EQUALS_INDEX compares the suffix to the image file extensions defined in imagetypes, and returns the index of the string that matches the URL suffix, if any.

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes")
```

If the URL in an incoming HTTP packet is http://www.example.com/mylogo.gif, the operator returns 4, which is the index of the pattern gif. You can further evaluate the index value by using operators that work with numbers. This is particularly useful when you have a large pattern set but want to perform a policy action only if the index has a particular value or lies within a specified range.

Following is an example of a compound expression that uses the GE(int) and LE(int) operators to determine whether a URL suffix is gif, tiff, or svg. If the compound expression evaluates to TRUE, an action is performed.

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(4) &&
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(6)
```

An expression that returns the index of a matched pattern can be used to define traffic subsets for a Web application. If you want to implement one set of NetScaler policies for BMP, JPEG, and PNG files, and a different set of policies for GIF, TIFF, and SVG files, you can use the pattern set imagetypes and define the following two content switching policies for a content switching virtual server:

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(3)
```

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(4)
```

For a complete list of the operators that are available for pattern sets and their descriptions, see Pattern Set Operators.

How String Matching with a Pattern Set Works

A pattern set contains a set of patterns, and each pattern is assigned a unique index. During policy evaluation, the operator compares the string that is identified in the packet with the patterns defined in the pattern set until a match is found. Then, depending on its function, the operator returns either a Boolean value that indicates whether or not a matching pattern was found or the index of the pattern that matches the string.

Following is an example pattern set, "imagetypes," that defines a set of image file extensions. The two example expressions that follow the definition of the pattern set demonstrate how the two types of operators work. The first example describes the functioning of an operator that returns a Boolean value. The second example describes the functioning of an operator that returns the index of the pattern that matches the string.

Table 1. Example Pattern Set "imagetypes"

Bound Pattern	Index
bmp	1
jpeg	2
png	3
gif	4
tiff	5
svg	6

Example 1. The following default syntax expression identifies the URL suffix in an HTTP packet. The operator EQUALS_ANY compares the suffix to the image file extensions defined in imagetypes and returns a Boolean value to indicate the result of the comparison.

```
HTTP.REQ.URL.SUFFIX.EQUALS_ANY("imagetypes")
```

If the URL in the incoming HTTP packet that the appliance is evaluating is http://www.example.com/homepageicon.jpeg, the operator returns a Boolean TRUE.

Example 2. The following default syntax expression identifies the URL suffix in an HTTP packet. The operator EQUALS_INDEX compares the suffix to the image file extensions defined in imagetypes, and returns the index of the string that matches the URL suffix, if any.

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes")
```

If the URL in an incoming HTTP packet is http://www.example.com/mylogo.gif, the operator returns 4, which is the index of the pattern gif. You can further evaluate the index value by using operators that work with numbers. This is particularly useful when you have a large pattern set but want to perform a policy action only if the index has a particular value or lies within a specified range.

Following is an example of a compound expression that uses the GE(int) and LE(int) operators to determine whether a URL suffix is gif, tiff, or svg. If the compound expression evaluates to TRUE, an action is performed.

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(4) &&
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(6)
```

An expression that returns the index of a matched pattern can be used to define traffic subsets for a Web application. If you want to implement one set of NetScaler policies for BMP, JPEG, and PNG files, and a different set of policies for GIF, TIFF, and SVG files, you can use the pattern set imagetypes and define the following two content switching policies for a content switching virtual server:

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(3)
```

```
HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(4)
```

For a complete list of the operators that are available for pattern sets and their descriptions, see Pattern Set Operators.

Configuring a Pattern Set

You configure a pattern set by specifying the strings that are to serve as patterns and binding them to the pattern set. Each pattern in the set has a unique index value. If you specify an index for the first pattern that you bind to a pattern set, you must specify an index for all the other patterns in the set, and you can modify the values at any time. If you do not specify an index for the first pattern, the NetScaler appliance assigns the pattern an index of value 1. Thereafter, the appliance assigns an index value to all the patterns that you bind to the set, and you cannot change them. After you have configured a pattern set for which the appliance has generated index values automatically, if you want to assign index values of your choice, you must create a new pattern set.

Index values are not regenerated automatically if one or more patterns are deleted or modified. For example, if the set contains five patterns, with indexes from 1 through 5, and if the pattern with an index of 3 is deleted, the other index values in the pattern set are not automatically regenerated to produce values from 1 through 4.

In the NetScaler command-line interface (CLI), you first create a pattern set by using the add policy patset command. Then, you create patterns and bind them to the pattern set, one pattern at a time, by using the bind policy patset command. In the NetScaler configuration utility, you perform all these tasks in a single dialog box.

Note: Pattern sets are case sensitive. Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."

To create a pattern set by using the NetScaler command line

At the NetScaler command prompt, type the following command to create a pattern set:

1. add policy patset <name>

Example

```
> add policy patset samplepatset
 Done
```

To bind a pattern to the pattern set by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a pattern to the pattern set and verify the configuration

- bind policy patset <name> <string> [-**index** <positive_integer>]
- show policy patset <name>

Example

```
bind policy patset samplepatset product1 -index 1
Done
show policy patset samplepatset
    Index is user-configured for the patset
1)   Bound Pattern:  product1       Index:  1
Done
>
```

To unbind a pattern from a pattern set by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a pattern from the pattern set and verify the configuration:

- unbind policy patset <name> <string>
- show policy patset <name>

Example

```
> unbind policy patset samplepatset product1
Done
> show policy patset samplepatset
 Done
>
```

To remove a pattern set by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a pattern set:

rm policy patset <name>

Example

```
> rm policy patset samplepatset
Done
>
```

Parameters for configuring a pattern set

name

The name of the pattern set. This is a required argument. Maximum length: 127 characters.

string

The pattern associated with the pattern set. This is a required argument. Maximum length: 255 characters.

Note: For pattern sets that are used in rewrite actions of type `REPLACE_ALL`, `DELETE_ALL`, `INSERT_AFTER_ALL`, and `INSERT_BEFORE_ALL`, the minimum length for the pattern is three characters.

index

The index of the pattern that is bound to the pattern set. This is an optional argument. Minimum value: 1. Maximum value: 4294967290.

To configure a pattern set by using the NetScaler configuration utility

1. In the navigation pane, expand **AppExpert**, and then click **Pattern Sets**.
2. In the details pane, do one of the following:
   - To create a pattern set, click **Add** to open the **Create Pattern Set** dialog box.
   - To modify an existing pattern set, select the pattern set, and then click **Open** to open the **Configure Pattern Set** dialog box.
3. If creating a pattern set, in the **Name** text box, type a name for the pattern set (maximum length, 127 characters).
4. Under **Specify Pattern**, type the first pattern and, optionally, specify values for the following parameters:
   - **Treat back slash as escape character**—Select this check box to specify that any backslash characters that you might include in the pattern are to be treated as escape characters.
   - **Index**—A user assigned index value, from 1 through 4294967290.
5. Verify that you have entered the correct characters, and then click **Add**.
6. Repeat steps 4 and 5 to add additional patterns, and then click **Create** or **OK**.

Using a Pattern Set

After you configure a pattern set, you can use it in a default syntax expression that passes the pattern set as an argument to an appropriate operator. For valid pattern set operators, see Pattern Set Operators.

Following is the format of the default syntax expression that compares a string with a pattern set:

<text>.<pattern set operator>("<name>")

In the preceding expression format, <text> represents any default syntax expression that identifies a string in a packet, such as `HTTP.REQ.BODY(uint).AFTER_REGEX(re).BEFORE_REGEX(re)`, and <name> is the name of the pattern set.

For example, if you want the appliance to determine whether the value of the Host header in a client request contains any of the patterns that are configured in a pattern set called "Patternset1," you can use the following default syntax expression:

HTTP.REQ.HEADER("Host").CONTAINS_ANY("Patternset1")

Note: You cannot use a pattern set in a classic expression.

Pattern Set Operators

The following table describes the operators that you can use with pattern sets. When you use an operator, replace `<text>` with the default syntax expression that identifies the string with which you want to perform string matching, and `replace <pattern_set_name>` with the name of the pattern set.

Table 1. Operators for Pattern Set

Operator	Description
`<text>.CONTAINS_ANY(<pattern_set_name>)`	Evaluates whether the target text contains any of the patterns that are bound to `<pattern_set_name>` and returns a Boolean TRUE if one or more matching patterns are found.
`<text>.SUBSTR_ANY(<pattern_set_name>)`	Selects the first string that matches any pattern in the given pattern set.
`<text>.EQUALS_ANY (<pattern_set_name>)`	Evaluates whether the target text exactly matches any of the patterns that are bound to `<pattern_set_name>` and returns a Boolean TRUE or FALSE to indicate the result of the evaluation.
`<text>.ENDSWITH_ANY(<pattern_set_name>)`	Evaluates whether the target text ends with any of the patterns that are bound to <pattern_set_name> and returns a Boolean TRUE or FALSE to indicate the result of the evaluation.

`<text>.STARTSWITH_ANY(<pattern_set_name>)`	Evaluates whether the target text starts with any of the patterns that are bound to `<pattern_set_name>` and returns a Boolean TRUE or FALSE to indicate the result of the evaluation.
`<text>.STARTSWITH_INDEX(<pattern_set_name>)`	Evaluates whether the target text starts with any of the patterns that are bound to `<pattern_set_name>` and, if a match is found, returns the numerical index of the matching pattern.
`<text>.ENDSWITH_INDEX(<pattern_set_name>)`	Evaluates whether the target text ends with any of the patterns that are bound to `<pattern_set_name>` and, if a match is found, returns the numerical index of the matching pattern.
`<text>.CONTAINS_INDEX(<pattern_set_name>)`	Evaluates whether the target text contains any of the patterns that are bound to `<pattern_set_name>` and, if a match is found, returns the numerical index of the matching pattern.
`<text>.EQUALS_INDEX(<pattern_set_name>)`	Evaluates whether the target text exactly matches any of the patterns that are bound to `<pattern_set_name>` and, if an exact match is found, returns the numerical index of the pattern.

Priority Queuing

The priority queuing feature lets you filter incoming HTTP traffic on the basis of categories that you create and define, and prioritize those HTTP requests accordingly. Priority queuing directs high-priority requests to the server ahead of low-priority requests, so that users who need resources for important business uses receive expedited access to your protected Web servers.

Note: The priority queuing feature is not supported in NetScaler 9.2 nCore.

To implement priority queuing, you create priority queuing policies that specify a priority, weight, threshold, and implicit action. When an incoming request matches a priority queuing policy, the request is processed as the associated action indicates. For example, you can create a priority queuing policy that places all matching requests above a certain threshold in a surge queue, while giving priority treatment to other requests.

You can bind up to three priority queuing policies to a single load balancing virtual server. The priority levels are:

Level 1

A Level 1 policy processes priority requests.

Level 2

A Level 2 policy processes requests that should receive responses as soon as Level 1 requests have been cleared from the queue.

Level 3

A Level 3 policy processes non-priority requests that receive responses only after requests in the first two queues have been cleared.

You can use weighted queuing to adjust the relative priority of each of these queues. Weights can range from 0 to 101. A weight of 101 tells the NetScaler appliance to clear all requests in that queue before forwarding any requests in the lower-priority queues to the Web server. A weight of 0 tells the appliance to send requests in that queue to the Web server only when there are no requests waiting in any of the other queues.

You must assign a unique name to each priority queuing policy. Policy names can be up to 127 characters. Multiple policies bound to the same load balancing virtual server cannot have the same priority level. No two virtual servers that have one or more common underlying physical services can have priority queuing configured or enabled on both virtual servers simultaneously.

To configure priority queuing the NetScaler, you perform the following steps:

- Enable the load balancing feature
- Define a server and service
- Define a load balancing virtual server

- Bind the service to the load balancing virtual server
- Enable the priority queuing feature
- Create the priority queuing policies
- Bind the priority queuing policies to the load balancing virtual server
- Enable priority queuing on load balancing virtual server

For information about enabling load balancing, creating servers, creating virtual servers and services, and binding these servers and services, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670. For complete information about policies and expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Enabling Priority Queuing

To use the priority queuing feature the NetScaler appliance, you must first enable it.

To enable priority queuing by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable priority queuing and verify the configuration:

- enable ns feature PriorityQueuing
- show ns feature

Example

```
> enable ns feature PriorityQueuing
 Done
> show ns feature

      Feature                 Acronym           Status
      -------                 -------           ------
1)    Web Logging             WL                ON
2)    Surge Protection        SP                OFF
3)    Load Balancing          LB                ON
.
.
.
8)    Priority Queuing        PQ                ON
.
23)   HTML Injection          HTMLInjection     ON
24)   NetScaler Push          push              OFF
Done
```

To enable priority queuing by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **Priority Queuing** check box.
4. Click **OK** and, in the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature is enabled.

Enabling Priority Queuing

To use the priority queuing feature the NetScaler appliance, you must first enable it.

To enable priority queuing by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable priority queuing and verify the configuration:

- enable ns feature PriorityQueuing
- show ns feature

Example

```
> enable ns feature PriorityQueuing
 Done
> show ns feature

      Feature                 Acronym           Status
      -------                 -------           ------
1)    Web Logging               WL              ON
2)    Surge Protection          SP              OFF
3)    Load Balancing            LB              ON
.
.
.
8)    Priority Queuing          PQ              ON
.
23)   HTML Injection            HTMLInjection   ON
24)   NetScaler Push            push            OFF
Done
```

To enable priority queuing by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **Priority Queuing** check box.
4. Click **OK** and, in the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature is enabled.

Configuring a Priority Queuing Policy

To configure a priority queuing policy, you can use either the configuration utility or the NetScaler command line.

Note: For more information about using the CLI commands, see the *Citrix NetScaler Command Reference Guide* at http://support.citrix.com/article/CTX128678.

To configure a priority queuing policy by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to configure a priority queuing policy and verify the configuration:

add pq policy <policyName> **-rule** <expression> **-priority** <positive_integer> [**-weight** <positive_integer>] [**-qDepth** <positive_integer> | **-polqDepth** <positive_integer>]

Example

```
> add pq policy pol_cgibin -rule "URL == '/cgi-bin/'" -priority 1
 Done
> show pq policy pol_cgibin
1)      Policy: pol_cgibin      Rule: URL == '/cgi-bin/'        Priority: 1     Weight: 10
        Hits: 0
 Done
```

Parameters for configuring a priority queuing policy

policyName

A name for your priority queuing policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the type of action.

rule

An expression that tells the policy which connections it should handle. For complete information about policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

priority

An integer from 1 to 3 that represents the priority queue that connections matching this policy should be placed in.

weight

An integer that represents the weight assigned to this priority queuing policy. For detailed information about policy weight, see Setting Up Weighted Queuing.

qDepth

An integer that represents the maximum number of connections that can be placed in the queue at one time.

polqDepth

An integer that represents the total number of waiting clients or requests belonging to the policy.

To configure a priority queuing policy by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then select **Priority Queuing**.
2. In the details pane, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the **policy**, and then click **Open**.
3. If you are creating a new policy, in the **Create PQ Policy** dialog box, in the **Name** text box, type a name for your new policy.

   If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.

4. In the **Rule** text box, either enter the policy expression directly, or click **New** to create a policy expression. If you click **New**, perform the following steps:
   a. In the **Create Expression** dialog box, click **Add.**
   b. In the **Add Expression** dialog box, in the **Flow Type** drop-down list, select a Flow Type. Your choices are **REQ** (for requests) and **RES** (for responses).
   c. In the **Protocol** drop-down list, select a protocol. If you selected **REQ** in the previous step, your choices are **HTTP** (Web-based connections), **SSL** (secure Web connections), **TCP** and **IP**. If you selected **RES** in the previous step, your choices are **HTTP**, **TCP** and **IP**.
   d. In the **Qualifier** drop-down list, select a qualifier.

      Your choices depend upon your selections in the previous step. Common choices are **HTTP VERSION** (the version of the HTTP connection), **HTTP HEADER** (the specified HTTP header), **TCP SOURCEPORT/ DESTPORT** (the source or destination port of a **TCP** connection), and **IP SOURCEIP/DESTIP** (the source or destination IP of the connection).

      If you choose **HTTP HEADER**, the **Header** text box appears beneath the original row of text boxes. You fill in the name of the HTTP header you want.

      For a complete description of the available choices, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

   e. In the **Operator** drop-down list, select an operator.

      For a complete description of the available choices, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

   f. In the **Value** text box, type the value you want to test for.

      This may be a text string or a number, depending upon the context. For a complete description of values appropriate to the specific context, see the *Citrix NetScaler*

Policy Configuration and Reference Guide at http://support.citrix.com/article/CTX128673.

g. Click **OK**. The expression is added in the **Expression** text box.

h. Click **Create**. The expression appears in the **Rule** text box.

5. In the **Priority** and **Weight** text boxes, type numeric values, for example, 1 and 30. For more information about **Priority** and **Weight**, see Setting Up Weighted Queuing.

6. Enter a numeric value for either **Queue Depth** or **Policy Queue Depth**, for example 234, and click **Create**.

   - **Queue Depth** Defines the total number of waiting clients or requests on the virtual server to which the policy is bound.

   - **Policy Queue Depth** Defines the total number of waiting clients or requests belonging to the policy.

   The policy is created and appears in the **Priority Queuing** page.

   Note: To create additional priority queuing policies, repeat the procedure in the preceding section, and click **Close** after you finish.

Binding a Priority Queuing Policy

After you create a priority queuing policy, you must bind it to the appropriate virtual server to put it into effect.

To bind a policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a policy and verify the configuration:

- bind lb vserver <name> **-policyName** <policyname>
- show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyname pol_cgibin
 Done
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+782 ms)
        Time since last state change: 26 days, 05:44:37.370
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:0

1)      Priority Queuing Policy : pol_cgibin
 Done
>
```

To bind a policy by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then select **Virtual Servers**.
2. In the details pane list of available virtual servers, select the virtual server to which you want to bind the priority queuing policy.
3. Click **Add**.
4. In the **Create Virtual Server** (Load Balancing) dialog box, click the **Policies** tab, and select the **Active** check box for the priority queuing policy that you created. A message appears in the status bar, stating that the policy has been configured successfully.

Setting Up Weighted Queuing

With priority queuing set, lower-priority requests are typically kept on hold while higher-priority requests are served. The lower-priority requests may therefore be delayed if there is a constant flow of higher-priority requests.

To prevent delays for low-priority requests across multiple priority levels, you can configure weighted queuing for serving requests. The default weights for the priorities are:

- Gold - Priority 1 - Weight 3
- Silver - Priority 2 - Weight 2
- Bronze - Priority 3 - Weight 1

You assign the minimum weight, zero (0), to requests that the NetScaler appliance should send to the server only if no requests are stored in any of the other queues. You assign the maximum weight, 101, to requests that the appliance should send to the server immediately, ahead of any requests stored in any of the other queues. Weights between these two set the relative priority of a particular queue in relation to the other queues. Queues with a higher weight are processed first; queues with a lower weight after the others have been processed. To assign the weights, see Configuring a Priority Queuing Policy.

Note: The weight assigned to a higher-priority queue must be larger than the weight assigned to a lower-priority queue. For example, the weight assigned to The Gold (Priority 1) queue must be greater than the weight assigned to the Silver (Priority 2) queue.

Rate Limiting

The rate limiting feature enables you to define the maximum load for a given network entity or virtual entity on the Citrix® NetScaler® appliance. The feature enables you to configure the appliance to monitor the rate of traffic associated with the entity and take preventive action, in real time, based on the traffic rate. This feature is particularly useful when the network is under attack from a hostile client that is sending the appliance a flood of requests. You can mitigate the risks that affect the availability of resources to clients, and you can improve the reliability of the network and the resources that the appliance manages.

You can monitor and control the rate of traffic that is associated with virtual and user-defined entities, including virtual servers, URLs, domains, and combinations of URLs and domains. You can throttle the rate of traffic if it is too high, base information caching on the traffic rate, and redirect traffic to a given load balancing virtual server if the traffic rate exceeds a predefined limit. You can apply rate-based monitoring to HTTP, TCP, and DNS requests.

To monitor the rate of traffic for a given scenario, you configure a *rate limit identifier*. A rate limit identifier specifies numeric thresholds such as the maximum number of requests or connections (of a particular type) that are permitted in a specified time period called a *time slice*.

Optionally, you can configure filters, known as *rate limit selectors*, and associate them with rate limit identifiers when you configure the identifiers. After you configure the optional limit selector and the limit identifier, you must invoke the limit identifier from an advanceda default syntax policy. You can invoke identifiers from any feature in which the identifier may be useful, including Rewrite, Responder, DNS, and Integrated Caching.

You can globally enable and disable SNMP traps for rate limit identifiers. Each trap contains cumulative data for the rate limit identifier's configured data collection interval (time slice), unless you specified multiple traps to be generated per time slice. For more information about configuring SNMP traps and managers, see the "SNMP" chapter in the *Citrix NetScaler Administration Guide* at http://support.citrix.com/article/CTX128667.

Configuring a Traffic Limit Selector

A traffic limit selector is an optional filter for identifying an entity for which you want to throttle access.The selector is applied to a request or a response and selects data points (keys) that can be analyzed by a rate limit identifier. These data points can be based on almost any characteristic of the traffic, including IP addresses, subnets, domain names, TCP or UDP identifiers, and particular strings or extensions in URLs.

A selector derives its functionality from a logical expression, called default syntax expression. For more information on default syntax expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

A limit selector consists of individual default syntax expressions called selectlets. Each selectlet is a non-compound default syntax expression. A traffic limit selector can contain up to five non-compound expressions called selectlets. Each selectlet is considered to be in an AND relationship with the other expressions. Following are some examples of selectlets:

http.req.url
http.res.body(1000>after_str(\"car_model\").before_str(\"made_in\")"
"client.ip.src.subnet(24)"

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the NetScaler considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the NetScaler, again, can count the same transaction more than once.

Note: If you modify an expression in a rate limiting selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a rate limiting selector named myLimitSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myLimitSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

To configure a traffic limit selector by using the NetScaler command line

At the NetScaler command line, type the following commands to configure a limit selector and verify the configuration:

- add ns limitSelector <selectorName> <rule>

- show ns limitSelector

Example

```
> add ns limitSelector client_dns_domain_and_source_ip client.udp.dns.domain client.ip.src
 Done
> show ns limitSelector client_dns_domain_and_source_ip
      Name: client_dns_domain_and_source_ip
      Expressions:
            1) client.udp.dns.domain
            2) client.ip.src
 Done
>
```

To modify or remove a limit selector by using the NetScaler command line

- To modify a limit selector, type the set ns limitSelector command, the name of the limit selector, and the selectlets, which must include the new selectlets that you want to add and any existing selectlets that you want to retain.
- To remove a limit selector, type the rm ns limitSelector command and the name of the limit selector.

Parameters for configuring a rate limit selector

selectorName

The name of rate limit selector. Maximum length: 31.

rule

A set of up to five non-compound default syntax expressions that are treated as one compound expression. Each atomic expression is treated as being in a logical AND relationship with the others in the selector definition.

To configure a traffic limit selector by using the NetScaler configuration utility

1. In the navigation pane, expand **AppExpert**, expand **Rate Limiting**, and then click **Limit Selectors**.

2. In the details pane, do one of the following:

   - To create a new limit selector, click **Add**.

   - To modify an existing limit selector, select the limit selector, and then click **Open**.

3. In the **Create Limit Selector** or **Configure Limit Selector** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a rate limit selector," as shown:

   - **Name***—selectorName

   - **Expressions**—rule

   * A required parameter

   Note: In the **Expressions** box, enter a valid parameter. For example, enter "http". Then, enter a period after this parameter. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. To select the next keyword in this expression prefix, double-click the selection in the drop-down menu. The **Expressions** text box displays both the first and second keywords for the expression prefix, for example, **"http.req"**. Continue adding expression components until the complete expression is formed.

4. Click **Add**.

5. Continue adding up to five non-compound expressions (selectlets).

6. Click **Create** or **OK**.

Configuring a Traffic Limit Selector

A traffic limit selector is an optional filter for identifying an entity for which you want to throttle access.The selector is applied to a request or a response and selects data points (keys) that can be analyzed by a rate limit identifier. These data points can be based on almost any characteristic of the traffic, including IP addresses, subnets, domain names, TCP or UDP identifiers, and particular strings or extensions in URLs.

A selector derives its functionality from a logical expression, called default syntax expression. For more information on default syntax expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

A limit selector consists of individual default syntax expressions called selectlets. Each selectlet is a non-compound default syntax expression. A traffic limit selector can contain up to five non-compound expressions called selectlets. Each selectlet is considered to be in an AND relationship with the other expressions. Following are some examples of selectlets:

http.req.url
http.res.body(1000>after_str(\"car_model\").before_str(\"made_in\")"
"client.ip.src.subnet(24)"

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the NetScaler considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the NetScaler, again, can count the same transaction more than once.

Note: If you modify an expression in a rate limiting selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a rate limiting selector named myLimitSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myLimitSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

To configure a traffic limit selector by using the NetScaler command line

At the NetScaler command line, type the following commands to configure a limit selector and verify the configuration:

- add ns limitSelector <selectorName> <rule>

- show ns limitSelector

Example

```
> add ns limitSelector client_dns_domain_and_source_ip client.udp.dns.domain client.ip.src
 Done
> show ns limitSelector client_dns_domain_and_source_ip
      Name: client_dns_domain_and_source_ip
      Expressions:
            1) client.udp.dns.domain
            2) client.ip.src
 Done
>
```

To modify or remove a limit selector by using the NetScaler command line

- To modify a limit selector, type the set ns limitSelector command, the name of the limit selector, and the selectlets, which must include the new selectlets that you want to add and any existing selectlets that you want to retain.
- To remove a limit selector, type the rm ns limitSelector command and the name of the limit selector.

Parameters for configuring a rate limit selector

selectorName

The name of rate limit selector. Maximum length: 31.

rule

A set of up to five non-compound default syntax expressions that are treated as one compound expression. Each atomic expression is treated as being in a logical AND relationship with the others in the selector definition.

To configure a traffic limit selector by using the NetScaler configuration utility

1. In the navigation pane, expand **AppExpert**, expand **Rate Limiting**, and then click **Limit Selectors**.

2. In the details pane, do one of the following:

   - To create a new limit selector, click **Add**.

   - To modify an existing limit selector, select the limit selector, and then click **Open**.

3. In the **Create Limit Selector** or **Configure Limit Selector** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a rate limit selector," as shown:

   - **Name***—selectorName

   - **Expressions**—rule

   * A required parameter

   Note: In the **Expressions** box, enter a valid parameter. For example, enter "http". Then, enter a period after this parameter. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. To select the next keyword in this expression prefix, double-click the selection in the drop-down menu. The **Expressions** text box displays both the first and second keywords for the expression prefix, for example, **"http.req"**. Continue adding expression components until the complete expression is formed.

4. Click **Add**.

5. Continue adding up to five non-compound expressions (selectlets).

6. Click **Create** or **OK**.

Configuring a Traffic Rate Limit Identifier

A rate limit identifier returns a Boolean TRUE if the amount of traffic exceeds a numeric limit within a particular time interval. The rate limit identifier definition can optionally include a limit selector. When you include a limit identifier in the compound default syntax expression in a policy rule, if you do not specify a limit selector, the limit identifier is applied to all the requests or responses that are identified by the compound expression.

To configure a traffic limit identifier from the NetScaler command line

At the NetScaler command line, type the following commands to configure a traffic limit identifier and verify the configuration:

- add ns limitIdentifier <limitIdentifier> [**-threshold** <positive_integer>] [**-timeSlice** <positive_integer>] [**-mode** <mode> [**-limitType** (BURSTY | SMOOTH)]] [**-selectorName** <string>] [**-maxBandwidth** <positive_integer>] [**-trapsInTimeSlice** <positive_integer>]

- show ns limitIdentifier

Example

```
> add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000 -mode REQUEST_RATE -limitType B
 Done
> show ns limitIdentifier 100_request_limit
Name: 100_request_limit
      Threshold:      100   Timeslice:     1000
      Traps :       30   Max Bandwidth:          0 kbps
      Selector: limit_100_requests_selector     Mode: REQUEST_RATE    Type: BURSTY
      Expressions:
            1) HTTP.REQ.METHOD
            2) CLIENT.IP.SRC
      Permit          100  Requests in  1000 ms
      Generate         30  Traps    in  1000 ms
      Hits: 0  Action Taken: 0
 Done
>
```

To modify or remove a limit identifier by using the NetScaler command line

- To modify a limit identifier, type the set ns limitIdentifier command, the name of the limit identifier, and the parameters to be changed, with their new values.
- To remove a limit identifier, type the rm ns limitIdentifier command and the name of the limit identifier.

Parameters for configuring a rate limit identifier

limitIdentifier

The name of rate limit identifier. Maximum length: 31.

threshold

Maximum number of requests in the time slice, specified as a positive integer. Used if you also configure a time slice interval. If this rate limit identifier uses the CONNECTIONS mode, the NetScaler limits the total number of connections that are made during the specified time interval to the specified threshold value.

Minimum: 1. Maximum: 4294967295. Default: 1.

timeSlice

A time interval, in multiples of 10 milliseconds, during which requests are tracked to see if they exceed the threshold. Used only when the mode is REQUEST_RATE.

Minimum: 10. Maximum: 42949672950. Default: 1000.

mode

What is tracked during the configured time slice. Possible values:

- REQUEST_RATE: Count HTTP, TCP, or DNS requests.
- CONNECTIONS: Count the active transactions on the client. This is useful for HTTP traffic (for example, Web crawlers that open many connections).
- NONE: Used if you want to only set the bandwidth.

Default: REQUEST_RATE.

limitType

Whether traffic is smooth or occurs in bursts when you configure the mode to be REQUEST_RATE:

- BURSTY: Most of the traffic arrives at one point in the interval (for example, the first 10 milliseconds (ms) of a one-second interval). An example of a bursty application is a Web page that may have a flurry of activity when a user accesses it, and then no

activity as the user reads the page.

- SMOOTH: An even distribution over the interval (for example, ten requests every ten milliseconds). If you choose SMOOTH, the NetScaler tracks requests at 10/ms intervals. For example, if you specify 4 requests per 200 ms, the NetScaler tracks at a rate of 1 request every 50 ms.

Possible values: BURSTY and SMOOTH. Default: BURSTY..

selectorName

The name of rate limit selector. Maximum Length: 31

maxBandwidth

Maximum bandwidth permitted, in kilobits (Kbps). The default value, 0, means that bandwidth is not used in this identifier.

Minimum value: 0. Maximum value: 4294967287. Default: 0.

trapsInTimeSlice

Number of SNMP traps to send in the time slice. A value of zero means that traps are disabled.

Default: 0. Minimum: 0. Maximum: 65535.

Traps are per identifier, not per identifier record. Use traps per time slice to increase the number of traps that are generated.

-trapsinTimeSlice positive_integer.

To configure a traffic limit identifier by using the NetScaler configuration utility

1. In the navigation pane, expand **AppExpert**, expand **Rate Limiting**, and then click **Limit Identifiers**.
2. In the details pane, do one of the following:
   - To create a new limit identifier, click **Add**.
   - To modify an existing limit identifier, select the limit identifier, and then click **Open**.
3. In the **Create Limit Identifier** or **Configure Limit Identifier** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a rate limit identifier," as shown:
   - **Name*****—limitIdentifier
   - **Selector**—selectorName (To create a new selector, click New next to the drop-down menu and define a selector as described in Configuring a Traffic Limit Selector.)
   - **Mode**—mode
   - **Limit Type**—limitType
   - **Threshold**—threshold
   - **Time Slice**—timeSlice
   - **Max Bandwidth**—maxBandwidth
   - **Traps**—trapsInTimeSlice

   * A required parameter
4. Click **Create** or **OK**.

Configuring and Binding a Traffic Rate Policy

You implement rate-based application behavior by configuring a policy in an appropriate NetScaler feature. The feature must support default syntax policies. The policy expression must contain the following expression prefix to enable the feature to analyze the traffic rate:

```
sys.check_limit(<limit_identifier>)
```

Where `limit_identifier` is the name of a limit identifier.

The policy expression must be a compound expression that contains at least two components:

- An expression that identifies traffic to which the rate limit identifier is applied. For example:

  ```
  http.req.url.contains("my_aspx.aspx").
  ```

- An expression that identifies a rate limit identifier, for example, `sys.check_limit("my_limit_identifier")`. This must be the last expression in the policy expression.

To configure a rate-based policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to configure a rate-based policy and verify the configuration:

add cache|dns|rewrite|responder policy <policy_name> **-rule** expression && sys.check_limit("<LimitIdentifierName>") [<feature-specific information>]

Following is a complete example of a rate-based policy rule. Note that this example assumes that you have configured the responder action, send_direct_url, that is associated with the policy. Note that the sys.check_limit parameter must be the last element of the policy expression:

add responder policy responder_threshold_policy "http.req.url.contains(\"myindex.html\") && sys.check_limi

For information about binding a policy globally or to a virtual server, see Binding Default Syntax Policies.

Rate Limiting Policy Parameters

Name

A name of up to 31 characters.

Expression

A default syntax expression that contains, at minimum, a component that identifies traffic to which the rate limit identifier should be applied and a sys.check_limit parameter.

Feature-specific information

Other required information for the policy definition, for example, actions or profiles to trigger if the policy evaluates to TRUE.

Note: You must specify sys.check_limit as the final expression element in the policy rule to ensure that the NetScaler updates the limit records only if the policy is true.

To configure a rate-based policy by using the NetScaler configuration utility

1. In the navigation pane, expand the feature in which you want to configure a policy (for example, **Integrated Caching**, **Rewrite**, or **Responder**), and then click **Policies**.
2. In the details pane, click **Add**. In **Name**, enter a unique name for the policy.
3. **Under Expression**, enter the policy rule, and make sure that you include the `sys.check_limit` parameter as the final component of the expression. For example:

   ```
   http.req.url.contains("my_aspx.aspx") && sys.check_limit("my_limit_identifier")
   ```

4. Enter feature-specific information about the policy.

   For example, you may be required to associate the policy with an action or a profile. For more information, see the feature-specific documentation.
5. Click **Create**, and then click **Close**.
6. Click **Save**.

Viewing the Traffic Rate

If traffic through one or more virtual servers matches a rate-based policy, you can view the rate of this traffic. The rate statistics are maintained in the limit identifier that you named in the rule for the rate-based policy. If more than one policy uses the same limit identifier, you can view the traffic rate as defined by hits to all of the policies that use the particular limit identifier.

To view the traffic rate by using the NetScaler command line

At the NetScaler command prompt, type: the following command to view the traffic rate:

show ns limitSessions <limitIdentifier>

Example

sh limitsession myLimitSession

Parameters for viewing the traffic rate

limitIdentifier

The name of the rate limit identifer. Maximum length: 31.

To view the traffic rate by using the NetScaler configuration utility

1. In the navigation pane, expand **AppExpert**, expand **Rate Limiting**, and then click **Limit Identifiers**.
2. Select a limit identifier whose traffic rate you want to view.
3. Click the **Show Sessions** button. If traffic through one or more virtual servers has matched a rate limiting policy that uses this limit identifier (and the hits are within the configured time slice for this identifier), the **Session Details** dialog box appears. Otherwise, you receive a "`No session exists`" message.

Testing a Rate-Based Policy

To test a rate-based policy, you can send traffic to any virtual server to which a rate-based policy is bound.

Task overview: Testing a rate-based policy

1. Configure a rate limit selector (optional) and a rate limit identifier (required). For example:

   ```
   add ns limitSelector sel_subnet Q.URL "CLIENT.IP.SRC.SUBNET(24)"
   add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode REQUEST_RATE  -limittype smooth  -s
   ```

2. Configure the action that you want to associate with the policy that uses the rate limit identifier. For example:

   ```
   add responder action resp_redirect redirect "\"http://response_site.com/\""
   ```

3. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier. For example, the policy can apply a rate limit identifier to all requests arriving from a particular subnet, as follows:

   ```
   add responder policy resp_subnet "SYS.CHECK_LIMIT(\"k_subnet\")" resp_redirect
   ```

4. Bind the policy globally or to a virtual server. For example:

   ```
   bind responder global resp_subnet 6 END -type DEFAULT
   ```

5. In a browser address bar, send a test HTTP query to a virtual server. For example:

   ```
   http://<IP of a vserver>/testsite/test.txt
   ```

6. At the NetScaler command prompt, type:

   ```
   show limitsession <limitIdentifier>
   ```

   Example

   ```
   > sh limitsession k_subnet
   1)      Time Remaining:       98 secs  Hits: 2                    Action Taken: 0
          Total Hash:    1718618  Hash String: /test.txt
          IPs gathered:
                1) 10.217.253.0
          Active Transactions: 0
    Done
   >
   ```

7. Repeat the query and check the limit identifier statistics again to verify that the statistics are being updated correctly.

Examples of Rate-Based Policies

The following table shows examples of rate-based policies.

Table 1. Examples of Rate-Based Policies

Purpose	Example
Limit the number of requests per second from a URL	add ns limitSelector ipLimitSelector http.req.url "client.ip.src" add ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000 -mode request_rate -limitType smooth -selectorName ipLimitSelector add responder action myWebSiteRedirectAction redirect "\"http://www.mycompany.com/\"" add responder policy ipLimitResponderPolicy "http.req.url.contains(\"m && sys.check_limit(\"ipLimitIdentifier\")" myWebSiteRedirectAction bind responder global ipLimitResponderPolicy 100 END -type default
Cache a response if the request URL rate exceeds 5 per 20000 milliseconds	add ns limitselector cacheRateLimitSelector http.req.url add ns limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice 2 -selectorName cacheRateLimitSelector add cache policy cacheRateLimitPolicy -rule "http.req.method.eq(get) && sys.check_limit(\"cacheRateLimitIdentifier\")" -action cache bind cache global cacheRateLimitPolicy -priority 10
Drop a connection on the basis of cookies received in requests from www.yourcompany.com if these requests exceed the rate limit	add ns limitSelector reqCookieLimitSelector "http.req.cookie .value(\"mycookie\")" "client.ip.src.subnet(24)" add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000 -selectorName reqCookieLimitSelector add responder action sendRedirectUrl redirect '\"http://www.mycompa + http.req.url' -bypassSafetyCheck YES add responder policy rateLimitCookiePolicy "http.req.url.contains(\"www.yourcompany.com\") && sys.check_limit(\"myLimitIdentifier\")" sendRedirectUrl

Drop a DNS packet if the requests from a particular client IP address and DNS domain exceed the rate limit	add ns limitSelector dropDNSRateSelector client.udp.dns.domain client add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode req -selectorName dropDNSRateSelector -maxBandwidth 1 -trapsintimeslice add dns policy dnsDropOnClientRatePolicy "sys.check_limit (\"dropDNSRateIdentifier\")" -drop yes
Limit the number of HTTP requests that arrive from the same subnet (with a subnet mask of 32) and that have the same destination IP address.	add ns limitSelector ipv6_sel "CLIENT.IPv6.src.subnet(32)" CLIENT.IPv6. add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName ipv6_sel add lb vserver ipv6_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTi add responder action redirect_page redirect "\"http://redirectpage.com add responder policy ipv6_resp_pol "SYS.CHECK_LIMIT(\"ipv6_id\")" redir bind responder global ipv6_resp_pol 5 END -type DEFAULT

Sample Use Cases for Rate-Based Policies

The following scenarios describe two uses of rate-based policies in global server load balancing (GSLB):

- The first scenario describes the use of a rate-based policy that sends traffic to a new data center if the rate of DNS requests exceeds 1000 per second.
- In the second scenario, if more than five DNS requests arrive for a local DNS (LDNS) client within a particular period, the additional requests are dropped.

Redirecting Traffic on the Basis of Traffic Rate

In this scenario, you configure a proximity-based load balancing method, and a rate-limiting policy that identifies DNS requests for a particular region. In the rate-limiting policy, you specify a threshold of 1000 DNS requests per second. A DNS policy applies the rate limiting policy to DNS requests for the region "Europe.GB.17.London.UK-East.ISP-UK." In the DNS policy, DNS requests that exceed the rate limiting threshold, starting with request 1001 and continuing to the end of the one-second interval, are to be forwarded to the IP addresses that are associated with the region "North America.US.TX.Dallas.US-East.ISP-US."

The following configuration demonstrates this scenario:

```
add ns limitSelector DNSSelector1 client.udp.dns.domain
add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1
add dns policy DNSLimitPolicy1 "client.ip.src.matches_location(\"Europe.GB.17.London.*.*\") &&
sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.*.*"
bind dns global DNSLimitPolicy1 5
```

Dropping DNS Requests on the Basis of Traffic Rate

In the following example of global server load balancing, you configure a rate limiting policy that permits a maximum of five DNS requests in a particular interval, per domain, to be directed to an LDNS client for resolution. Any requests that exceed this rate are dropped. This type of policy can help protect the NetScaler from resource exploitation. For example, in this scenario, if the time to live (TTL) for a connection is five seconds, this policy prevents the LDNS from requerying a domain. Instead, it uses data that is cached on the NetScaler.

```
add ns limitSelector LDNSSelector1 client.udp.dns.domain client.ip.src
add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName LDNSSelector1
add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(\".\") && sys.check_limit(\"LDNSLimitIdentifier1\
bind dns global LDNSPolicy1 6
show gslb vserver gvip
gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1)      site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2)      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3)      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
1)      www.gslbindia.com       TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
```

Redirecting Traffic on the Basis of Traffic Rate

In this scenario, you configure a proximity-based load balancing method, and a rate-limiting policy that identifies DNS requests for a particular region. In the rate-limiting policy, you specify a threshold of 1000 DNS requests per second. A DNS policy applies the rate limiting policy to DNS requests for the region "Europe.GB.17.London.UK-East.ISP-UK." In the DNS policy, DNS requests that exceed the rate limiting threshold, starting with request 1001 and continuing to the end of the one-second interval, are to be forwarded to the IP addresses that are associated with the region "North America.US.TX.Dallas.US-East.ISP-US."

The following configuration demonstrates this scenario:

```
add ns limitSelector DNSSelector1 client.udp.dns.domain
add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1
add dns policy DNSLimitPolicy1 "client.ip.src.matches_location(\"Europe.GB.17.London.*.*\") &&
sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.*.*"
bind dns global DNSLimitPolicy1 5
```

Dropping DNS Requests on the Basis of Traffic Rate

In the following example of global server load balancing, you configure a rate limiting policy that permits a maximum of five DNS requests in a particular interval, per domain, to be directed to an LDNS client for resolution. Any requests that exceed this rate are dropped. This type of policy can help protect the NetScaler from resource exploitation. For example, in this scenario, if the time to live (TTL) for a connection is five seconds, this policy prevents the LDNS from requerying a domain. Instead, it uses data that is cached on the NetScaler.

```
add ns limitSelector LDNSSelector1 client.udp.dns.domain client.ip.src
add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName LDNSSelector1
add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(\".\") && sys.check_limit(\"LDNSLimitIdentifier1\
bind dns global LDNSPolicy1 6
show gslb vserver gvip
gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1)      site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2)      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3)      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
1)      www.gslbindia.com       TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
```

Responder

Today's complex Web configurations often require different responses to HTTP requests that appear, on the surface, to be similar. When users request a Web site's home page, you may want to provide a different home page depending on where each user is located, which browser the user is using, or which language(s) the browser accepts and the order of preference. You might want to break the connection immediately if the request is coming from an IP range that has been generating DDoS attacks or initiating hacking attempts.

With the **Responder** feature, responses can be based on who sends the request, where it is sent from, and other criteria with security and system management implications. The feature is simple and quick to use. By avoiding the invocation of more complex features, it reduces CPU cycles and time spent in handling requests that do not require complex processing.

For handling sensitive data such as financial information, if you want to ensure that the client uses a secure connection to browse a site, you can redirect the request to secure connection by using https:// instead of http://.

To use the Responder feature, do the following;

- Enable the **Responder** feature on the NetScaler.
- Configure responder actions. The action can be to generate a custom response, redirect a request to a different Web page, or reset a connection.
- Configure responder policies. The policy determines the requests (traffic) on which an action has to be taken.
- Bind each policy to a bind point put it into effect. A bind point refers to an entity at which NetScaler examines the traffic to see if it matches a policy. For example, a bind point can be a load balancing virtual server.

You can specify a default action for requests that do not match any policy, and you can bypass the safety check for actions that would otherwise generate error messages.

The Rewrite feature of NetScaler helps in rewriting some information in the requests or responses handled by NetScaler. The following section shows some differences between the two features.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a http request to new Web sites or Web pages
- Responding with some custom response

- Dropping or resetting a connection at request level

In case of a responder policy, the NetScaler examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

In case of a rewrite policy, the NetScaler examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the NetScaler to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

Enabling the Responder Feature

To use the Responder feature, you must first enable it.

To enable the responder feature by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable the responder feature and verify the configuration:

- enable ns feature Responder
- show ns feature

Example

```
enable ns feature Responder
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging             WL               ON
 2)   Surge Protection        SP               ON
 .
 .
 .
 22)  Responder               RESPONDER        ON
 23)  HTML Injection          HTMLInjection    ON
 24)  NetScaler Push          push             OFF
 Done
>
```

To enable the responder feature by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **Responder** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, click **YES**. A message appears in the status bar, stating that the feature has been enabled.

Enabling the Responder Feature

To use the Responder feature, you must first enable it.

To enable the responder feature by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable the responder feature and verify the configuration:

- enable ns feature Responder
- show ns feature

Example

```
enable ns feature Responder
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging             WL               ON
 2)   Surge Protection        SP               ON
 .
 .
 .
 22)  Responder               RESPONDER        ON
 23)  HTML Injection          HTMLInjection    ON
 24)  NetScaler Push          push             OFF
 Done
>
```

To enable the responder feature by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **Responder** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, click **YES**. A message appears in the status bar, stating that the feature has been enabled.

Configuring a Responder Action

After enabling the responder feature, you must configure one or more actions for handling requests. The responder supports two types of actions:

Respondwith Action

Sends the designated response without forwarding the request to a Web server. Instead, the NetScaler appliance substitutes for and acts as a Web server itself.

Redirect Action

Redirects the request to a different Web page or Web server. A Redirect action can redirect requests originally sent to a “dummy” Web site that exists in DNS, but for which there is no actual Web server, to an actual Web site. It can also redirect search requests to an appropriate URL.

Normally the text for a Respondswith action consists of a Web server error code and brief HTML page. Normally, the redirection target for a Redirect action consists of a complete URL.

To configure a responder action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a responder action and verify the configuration:

- add responder action <name> <type> <target> [**-bypassSafetyCheck** (YES | NO)]
- show responder action

Example

For example, to create a responder action that displays a “Not Found” error page for URLs that do not exist, you would type the following:

```
add responder action act404Error respondWith "'HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL
 Done
> show responder action

1)     Name: act404Error
       Operation: respondwith
       Target: "HTTP/1.1 404 Not Found

"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."
       BypassSafetyCheck : NO
```

```
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
Done
```

To modify an existing responder action by using the NetScaler command line

At the NetScaler command prompt, type the following command to modify an existing responder action and verify the configuration:

- set responder action <name> **-target** <string> [**-bypassSafetyCheck** (YES | NO)]
- show responder action

Example

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL_SAF
 Done
> show responder action

1)      Name: act404Error
        Operation: respondwith
        Target: "HTTP/1.1 404 Not Found

"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."
        BypassSafetyCheck : NO
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done
```

To remove a responder action by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a responder action and verify the configuration:

- rm responder action <name>
- show responder action

Example

```
rm responder action act404Error
 Done

> show responder action
 Done
```

Parameters for configuring a responder action

name

A name for your new action, or the name of the existing action you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

type

The type of responder action. Possible values: RESPONDWITH (send response specified by target), REDIRECT (redirect the request to the URL specified by target).

target

The HTTP string to be sent as a response, or URL to which the request is redirected. Must consist of one or more strings enclosed in straight double quotes, with the entire response enclosed in straight single quotes. Within a response, type a plus sign (+) between separate double-quoted strings. Type \r\n to begin a new line.

bypassSafetyCheck

Whether to bypass the appliance's built-in safety checks when adding or modifying this action. Possible values are YES and NO.

To configure a responder action by using the configuration utility

1. In the navigation pane, expand **Responder**, and then click **Actions**.
2. In the details pane, do one of the following:
   - To create a new action, click **Add**.
   - To modify an existing action, select the action, and then click **Open**.
3. In the **Add Responder Action** or **Configure Responder Action** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring a responder action" as follows (asterisk indicates a required parameter):
   - Name*—name (Cannot be changed for a previously configured action)
   - Type*—type
   - Target*—target
   - Bypass Safety Check—byPassSafetyCheck

   If you want help creating the target for a new action, while the cursor is in the **Target** text box you can either hold down the **Control** key and press the **space bar**, or you can use the **Add Expression** dialog box as described in **"**To add an expression by using the **Add Expression** dialog box**"** below.
4. Click **Create** or **OK**, depending on whether you are creating a new action or modifying an existing action.
5. Click **Close**. A message appears in the status bar, stating that the feature has been enabled.
6. To delete a responder action, select the action, and then click **Remove**. A message appears in the status bar, stating that the feature has been disabled.

To add an expression by using the Add Expression dialog box

1. In the **Create Responder Action** or **Configure Responder Action** dialog box, click **Add**.
2. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.

   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   SYS

   The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   CLIENT

   The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the **Help** window below the **Construct Expression** window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished. For more information about the PI expressions language and creating expressions for responder policies, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Configuring a Responder Policy

After you configure a responder action, you must next configure a responder policy to select the requests to which the NetScaler appliance should respond. A responder policy is based on a rule, which consists of one or more expressions. The rule is associated with an action, which is performed if a request matches the rule.

Note: For creating and managing responder policies, the configuration utility provides assistance that is not available at the NetScaler command line.

To configure a responder policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to add a new responder policy and verify the configuration:

- add responder policy <name> <expression> <action> [<undefaction>]
- show responder policy <name>

Example

```
> add responder policy policyThree "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)" RESET
 Done
> show responder policy policyThree

      Name: policyThree
      Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
      Responder Action: RESET
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
 Done
```

To modify an existing responder policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to modify an existing responder policy and verify the configuration:

- set responder policy <name> [**-rule** <expression>] [**-action** <string>] [**-undefAction** <string>]

- show responder policy <name>

To remove a responder policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a responder policy and verify the configuration:

- rm responder policy <name>
- show responder policy

Example

```
>rm responder policy pol404Error
 Done

> show responder policy
 Done
```

Parameters for configuring a responder policy

name

A name for the policy, or the name of the existing policy you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of request this policy was configured to match.

rule

The expression that defines the rule for this policy. The expression can be a simple expression or a complex expression that contains several expressions in structured relationship to one another. Expressions are written in the NetScaler Policy Infrastructure (PI) language. For more information about PI, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

action

The name of the responder action associated with the policy. You can choose either the built-in ‘NOOP’ or ‘RESET’ actions, or a responder action you have configured.

undefaction

The action to use if the policy generates an UNDEF event. You can select either the NOOP, RESET, or DROP action, or configure the NetScaler appliance use the configured global undefined action.

Any responder-specific undefined action you configure will override the global undefined action.

To configure a responder policy by using the configuration utility

1. In the navigation pane, expand **Responder**, and then click **Policies**.
2. In the details pane, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Open**.
3. In the **Create Responder Policy** or **Configure Responder Policy** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring a responder policy" as follows (asterisk indicates a required parameter):
   - Name*—name (Cannot be changed for a previously configured policy.)
   - Action*—action
   - Expression*—expression
   - Undefined-Result Action—undefaction

   If you want help creating an expression for a new policy, while your cursor is in the **Expression** text box you can either hold down the **Control** key and press the **space bar**, or you can use the **Add Expression** dialog box as described in “To add an expression by using the **Add Expression** dialog box.”
4. Click **Create** or **OK**, depending on whether you are creating a new policy or modifying an existing policy.
5. Click **Close**. A message appears in the status bar, stating that the feature has been configured.

Binding a Responder Policy

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler, or to a specific virtual server, so that the policy applies only to requests whose destination IP address is the VIP of that virtual server.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The responder feature implements only the first policy that a request matches, not any additional policies that it might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Note: Responder policies cannot be bound to TCP-based virtual servers.

To globally bind a responder policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to globally bind a responder policy and verify the configuration:

- bind responder global <policyName> <priority> [<gotoPriorityExpression [**-type** <type>] [**-invoke** (<labelType> <labelName>)]
- show responder global

Example

```
> bind responder global poliError 100
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
       Number of bound policies: 1
```

Done

To bind responder policy to a specific virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following command to bind responder policy to a specific virtual server and verify the configuration:

bind lb vserver <name> **-policyname** <policy_name> **-priority** <priority>

Example

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
1)      vs-loadbal (10.102.29.20:80) - HTTP      Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
2)      vs-cont-sw (0.0.0.0:0) - TCP     Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
        Time since last state change: 2 days, 00:00:04.260
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
 Done
```

Parameters for binding a responder policy

name

The name of the virtual server to which you want to bind this policy.

policyname

The name of the responder policy you want to bind.

priority

The priority assigned to this responder policy.

type

Bindpoint, specifying where to bind the policy.

invoke

Invoke flag.

labelType

Type of policy label invocation.

labelName

Name of the label to invoke if the current policy rule evaluates to TRUE.

To globally bind a responder policy by using the configuration utility

1. In the navigation pane, expand **Responder**, and then click **Policies**.
2. On the **Responder Policies** page, select a responder policy, and then click **Policy Manager**.
3. In the **Responder Policy Manager** dialog box **Bind Points** menu, select **Default Global**.
4. Click **Insert Policy** to insert a new row and display a drop-down list of all unbound responder policies.
5. Click one of the policies on the list. That policy is inserted into the list of globally bound responder policies.
6. Click **Apply Changes**.
7. Click **Close**. A message appears in the status bar, stating that the configuration has been successfully completed.

To bind a responder policy to a specific virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers.**
2. On the **Load Balancing Virtual Servers** page, select the virtual server to which you want to bind the responder policy, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, select the **Policies** tab, which displays a list of all policies configured on your NetScaler appliance.
4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click **OK**. A message appears in the status bar, stating that the configuration has been successfully completed.

Setting the Responder Default Action

The NetScaler appliance generates an undefined event (UNDEF event) when a request does not match a responder policy, and then carries out the default action assigned to undefined events. By default, that action is to forward the request to the next feature without changing it. This default behavior is normally what you want; it ensures that requests that do not require special handling by a specific responder action are sent to your Web servers and clients receive access to the content that they requested.

If the Web site(s) your NetScaler appliance protects receive a significant number of invalid or malicious requests, however, you may want to change the default action to either reset the client connection or drop the request. In this type of configuration, you would write one or more responder policies that would match any legitimate requests, and simply redirect those requests to their original destinations. Your NetScaler appliance would then block any other requests as specified by the default action you configured.

You can assign any one of the following actions to an undefined event:

NOOP

The NOOP action aborts responder processing but does not alter the packet flow. This means that the appliance continues to process requests that do not match any responder policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers and is the default setting.

RESET

If the undefined action is set to RESET, the appliance resets the client connection, informing the client that it must re-establish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

DROP

If the undefined action is set to DROP, the appliance silently drops the request without responding to the client in any way. This action is appropriate for requests that appear to be part of a DDoS attack or other sustained attack on your servers.

Note: UNDEF events are triggered only for client requests. No UNDEF events are triggered for responses.

To set the undefined action by using the NetScaler command line

At the NetScaler command prompt, type the following command to set the undefined action and verify the configuration:

- set responder param **-undefAction** (RESET|DROP|NOOP)
- show responder param

Example

```
>set responder param -undefAction RESET
 Done
> show responder param
       Action Name: RESET
 Done
>
```

To set the undefined action by using the configuration utility

1. In the navigation pane, expand **Responder**, and then under **Settings**, click the **Change Responder Settings** link.
2. In the **Set Responder Params** dialog box, under **Global Undefined-Result Action**, select **NOOP**, **RESET**, or **DROP**.
3. Click **OK**. A message appears in the status bar, stating that the Responder Parameters have been configured.

Responder Action and Policy Examples

Responder actions and policies are powerful and complex, but you can get started with relatively simple applications. For typical examples, see Example: Blocking Access from Specified IPs and Example: Redirecting a Client to a new URL.

Example: Blocking Access from Specified IPs

The following procedures block access to your protected Web site(s) by clients originating from the CIDR 222.222.0.0/16. The responder sends an error message stating that the client is not authorized to access the URL requested.

To block access by using the NetScaler command line

At the NetScaler command prompt, type the following commands to block access:

- add responder action act_unauthorized respondwith '"HTTP/1.1 200 OK\r\n\r\n" + "Client: " + CLIENT.IP.SRC + " is not authorized to access URL:" + HTTP.REQ.URL.HTTP_URL_SAFE'
- add responder policy pol_un "CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)" act_un
- bind responder global pol_un 10

To block access by using the configuration utility

1. In the navigation pane, expand **Responder**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, do the following:
   a. In the **Name** text box, type `act_unauthorized`.
   b. Under **Type**, select **Respond with**.
   c. In the **Target** text area, type the following string:
   ```
   "HTTP/1.1 200 OK\r\n\r\n" + "Client: " + CLIENT.IP.SRC + " is not authorized to access URL:" + HTTP.REQ.URL.HTTP_URL_SAFE
   ```
   d. Click **Create**, and then click **Close**.

   The responder action you configured, named `act_unauthorized`, now appears in the **Responder Actions** page.
4. In the navigation pane, click **Policies**.
5. In the details pane, click **Add**.
6. In the **Create Responder Policy** dialog box, do the following:
   a. In the **Name** text box, type `pol_unauthorized`.
   b. Under **Action**, select **act_unauthorized**.
   c. In the **Expression** window, type the following rule:
   ```
   CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
   ```
   d. Click **Create**, then click **Close**.

   The responder policy you configured, named **pol_unauthorized**, now appears in the **Responder Policies** page.
7. Globally bind your new policy, `pol_unauthorized`, as described in Binding a Responder Policy.

Example: Redirecting a Client to a new URL

The following procedures redirect clients who access your protected Web site(s) from within the CIDR 222.222.0.0/16 to a specified URL.

To redirect clients by using the NetScaler command line

At the NetScaler command prompt, type the following commands to redirect clients and verify the configuration:

- add responder action act_redirect redirect "http://www.example.com/404.html"
- show responder action act_redirect

- add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)" act_redirect
- show responder policy pol_redirect
- bind responder global pol_redirect 10

Example

```
>add responder action act_redirect redirect "' http ://www.example.com/404.html "'
 Done
> show responder action act_redirect

1)      Name: act_redirect
       Operation: redirect
       Target: " http ://www.example.com/404.html "
       BypassSafetyCheck : NO
       Hits: 0
       Undef Hits: 0
       Action Reference Count: 0
 Done
>


>add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)"act_redirect
 Done
> show responder policy pol_redirect

       Name: pol_redirect
       Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
       Responder Action: act_redirect
       UndefAction: Use Global
       Hits: 0
       Undef Hits: 0
 Done
>
```

To redirect clients by using the configuration utility

1. In the navigation pane, expand **Responder**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, do the following:
   a. In the **Name** text box, type `act_redirect`.
   b. Under **Type**, select **Redirect**.
   c. In the **Target** text area, type the following string:
   `http://www.example.com/404.html`
   d. Click **Create**, then click **Close**.

   The responder action you configured, named **act_redirect**, now appears in the **Responder Actions** page.
4. In the navigation pane, click **Policies**.
5. In the details pane, click **Add**.
6. In the **Create Responder Policy** dialog box, do the following:
   a. In the **Name** text box, type `pol_redirect`.
   b. Under **Action**, select **act_redirect**.
   c. In the **Expression** window, type the following rule:`CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)`
   d. Click **Create**, then click **Close**.

   The responder policy you configured, named pol_redirect, now appears in the **Responder Policies** page.
7. Globally bind your new policy, pol_redirect, as described in Binding a Responder Policy.

Rewrite

Rewrite refers to the rewriting of some information in the requests or responses handled by the NetScaler appliance. Rewriting can help in providing access to the requested content without exposing unnecessary details about the Web site's actual configuration. A few situations in which the rewrite feature is useful are described below:

- To improve security, the NetScaler can rewrite all the http:// links to https:// in the response body.
- In the SSL offload deployment, the insecure links in the response have to be converted into secure links. Using the rewrite option, you can rewrite all the *http://* links to *https://* for making sure that the outgoing responses from NetScaler to the client have the secured links.
- If a Web site has to show an error page, you can show a custom error page instead of the default 404 Error page. For example, if you show the home page or site map of the Web site instead of an error page, the visitor remains on the site instead of moving away from the Web site.
- If you want to launch a new Web site, but use the old URL, you can use the **Rewrite** option.
- When a topic in a site has a complicated URL, you can rewrite it with a simple, easy-to-remember URL (also referred to as 'cool URL').
- You can append the default page name to the URL of a Web site. For example, if the default page of a company's Web site is '*http://www.abc.com/index.php*', when the user types '*abc.com*' in the address bar of the browser, you can rewrite the URL to '*abc.com/index.php*'.

When you enable the rewrite feature, NetScaler can modify the headers and body of HTTP requests and responses.

To rewrite HTTP requests and responses, you can use protocol-aware NetScaler policy expressions in the rewrite policies you configure. The virtual servers that manage the HTTP requests and responses must be of type HTTP or SSL. In HTTP traffic, you can take the following actions:

- Modify the URL of a request
- Add, modify or delete headers
- Add, replace, or delete any specific string within the body or headers.

To rewrite TCP payloads, consider the payload as a raw stream of bytes. Each of the virtual servers that managing the TCP connections must be of type TCP or SSL_TCP. The term *TCP rewrite* is used to refer to the rewrite of TCP payloads that are not HTTP data. In TCP traffic, you can add, modify, or delete any part of the TCP payload.

For examples to use the rewrite feature, see Rewrite Action and Policy Examples

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a http request to new Web sites or Web pages
- Responding with some custom response
- Dropping or resetting a connection at request level

In case of a responder policy, the NetScaler examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

In case of a rewrite policy, the NetScaler examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the NetScaler to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

How Rewrite Works

After you configure rewrite, when the NetScaler receives a request or sends a response, it checks to see if there are any rewrite policies configured. If it finds any rewrite policies, it tests the request or response against those policies, and adds the action associated with each matching policy to a list of actions to be performed. (A match occurs when the characteristics specified in the policy match the characteristics of the request or response being evaluated.) When it has finished evaluating rewrite policies, it performs the actions before passing the request or response on to the Web server or client computer.

The following figure illustrates how the rewrite feature processes each individual request or response.

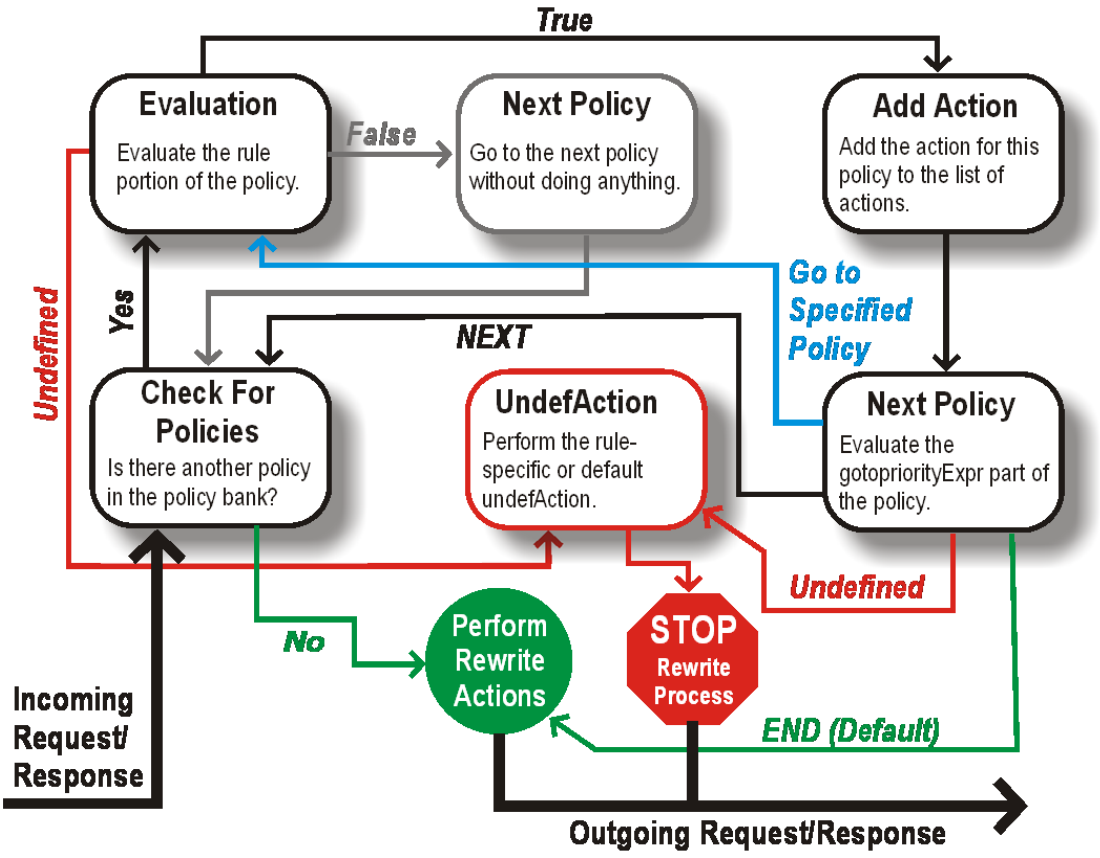


Figure 1. The Rewrite Process

After you configure your rewrite actions and associated rewrite policies, and bind the policies to the proper bind points, the NetScaler begins to evaluate requests and responses against these policies. When a user's browser sends an HTTP request to your Web server, the NetScaler checks its list of policies. If it finds rewrite policies, it evaluates each policy in order of priority.

The policies with the lowest priority numbers are evaluated first. Unlike other policy based features, however, the rewrite feature does not cease to evaluate rewrite policies once it finds a match. Instead, it continues to evaluate rewrite policies until it reaches the end of

the policy list. This means that the rewrite feature evaluates all policies against each incoming request or response for a match, except in the following cases:

- A policy evaluates as TRUE (or matches the request or response), and the gotoPriorityExpr for that policy points to a specific policy, bypassing intervening policies.
- A policy matches, and the gotoPriorityExpr for that policy is set to END, directing the rewrite feature to stop evaluating policies and apply the current list of actions to the request or response.
- A policy evaluates as UNDEFINED (or generates an internal error). In this case, the rewrite feature performs the action assigned to the UNDEFINED condition (called the undefAction) and discards the current actions list.

Only after it has finished evaluating policies does the rewrite feature perform the actions identified by the matching policies. It then performs each action on the unmodified request or response, rather than performing each action on the request or response as modified by any previous actions.

This has two important implications. First, the results of performing a series of rewrite actions will be the same regardless of the order in which those actions are performed. Second, the results of performing a series of rewrite actions on the same HTTP header or same portion of the HTTP body are undefined; almost anything can happen if you attempt this. For that reason, you should not configure two rewrite actions that affect the same HTTP header or the same part of the HTTP body.

How Rewrite Works

After you configure rewrite, when the NetScaler receives a request or sends a response, it checks to see if there are any rewrite policies configured. If it finds any rewrite policies, it tests the request or response against those policies, and adds the action associated with each matching policy to a list of actions to be performed. (A match occurs when the characteristics specified in the policy match the characteristics of the request or response being evaluated.) When it has finished evaluating rewrite policies, it performs the actions before passing the request or response on to the Web server or client computer.

The following figure illustrates how the rewrite feature processes each individual request or response.

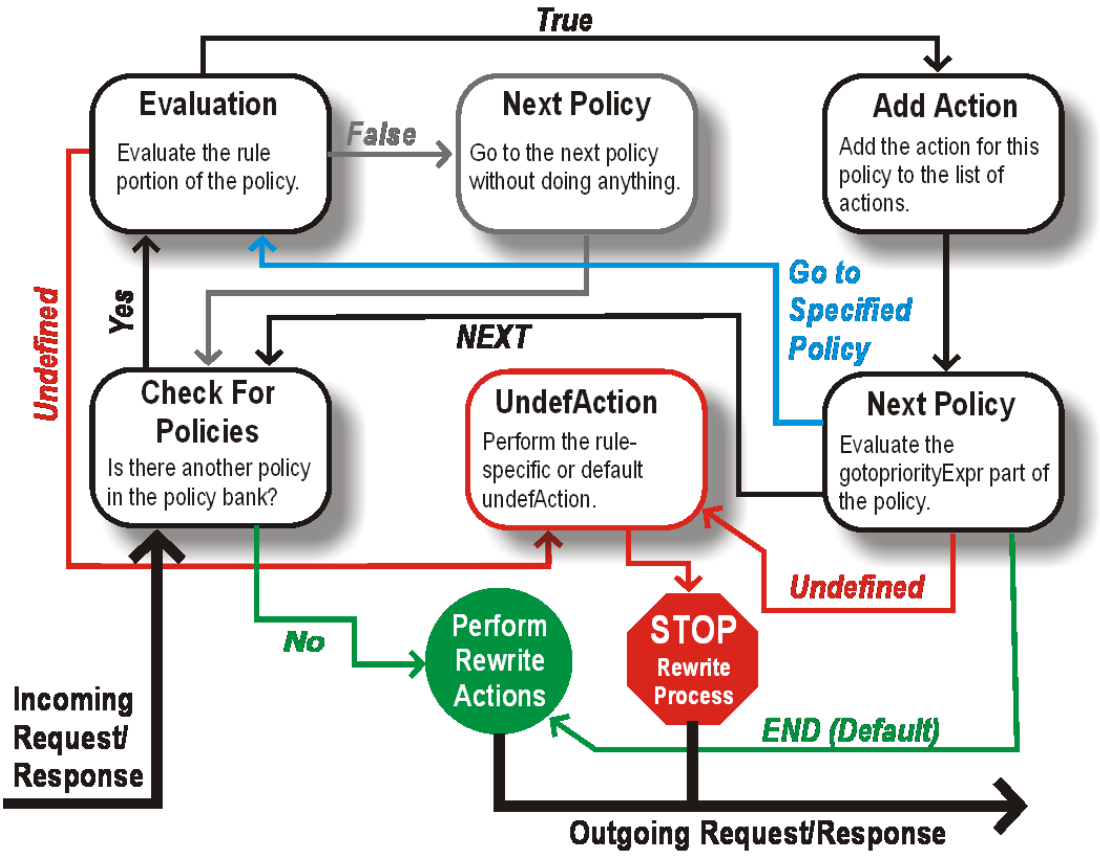


Figure 1. The Rewrite Process

After you configure your rewrite actions and associated rewrite policies, and bind the policies to the proper bind points, the NetScaler begins to evaluate requests and responses against these policies. When a user's browser sends an HTTP request to your Web server, the NetScaler checks its list of policies. If it finds rewrite policies, it evaluates each policy in order of priority.

The policies with the lowest priority numbers are evaluated first. Unlike other policy based features, however, the rewrite feature does not cease to evaluate rewrite policies once it finds a match. Instead, it continues to evaluate rewrite policies until it reaches the end of

the policy list. This means that the rewrite feature evaluates all policies against each incoming request or response for a match, except in the following cases:

- A policy evaluates as TRUE (or matches the request or response), and the gotoPriorityExpr for that policy points to a specific policy, bypassing intervening policies.
- A policy matches, and the gotoPriorityExpr for that policy is set to END, directing the rewrite feature to stop evaluating policies and apply the current list of actions to the request or response.
- A policy evaluates as UNDEFINED (or generates an internal error). In this case, the rewrite feature performs the action assigned to the UNDEFINED condition (called the undefAction) and discards the current actions list.

Only after it has finished evaluating policies does the rewrite feature perform the actions identified by the matching policies. It then performs each action on the unmodified request or response, rather than performing each action on the request or response as modified by any previous actions.

This has two important implications. First, the results of performing a series of rewrite actions will be the same regardless of the order in which those actions are performed. Second, the results of performing a series of rewrite actions on the same HTTP header or same portion of the HTTP body are undefined; almost anything can happen if you attempt this. For that reason, you should not configure two rewrite actions that affect the same HTTP header or the same part of the HTTP body.

Enabling the Rewrite Feature

Before you can use rewrite, you must enable the rewrite feature.

To enable the rewrite feature by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable the rewrite feature and verify the configuration:

- enable ns feature REWRITE
- show ns feature

Example

```
> enable ns feature REWRITE
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               ON
.
.
.
19)   Rewrite                  REWRITE          ON
.
.
24)   NetScaler Push           push             OFF
Done
```

To enable the rewrite feature by using the configuration utility

1. In the navigation pane, click **System**, and then click **Settings**.
2. In the details pane, under **Modes & Features**, click **Change Basic Features**.
3. In the **Configure Basic Features** dialog box, select the **Rewrite** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature was enabled.

Configuring a Rewrite Action

After enabling the rewrite feature, you need to configure one or more actions unless a built-in rewrite action is sufficient. All of the built-in actions have names beginning with the string ns_cvpn, followed by a string of letters and underscore characters. Built-in actions perform useful and complex tasks such as decoding parts of a clientless VPN request or response or modifying JavaScript or XML data. The built-in actions can be viewed, enabled, and disabled, but cannot be modified or deleted.

User-configured rewrite actions can modify either the HTTP headers of a request or response, or the HTTP body of a response. To configure a rewrite action, you assign it a name, specify an action type, and add one or more arguments specifying additional data. The following table describes the action types and the arguments you use with them.

Table 1. Rewrite Action Types and Their Arguments

Rewrite Action Type	Argument 1	Argument 2
INSERT_HTTP_HEADER: Inserts the HTTP header you specify into the HTTP request or response. This is the default choice.	The HTTP header you want to insert. For example, if you want to insert the client IP from which a request is sent, type `Client-IP`.	A string expression that describes the contents of the header you want to insert. For example, if you want to insert the Client IP from which a request is sent, type `CLIENT.IP.SRC`.
INSERT_BEFORE: Inserts a new string before the designated string in the HTTP header or body.	A string expression that describes the string before which you want to insert a new string. For example, if you want to find the hostname www.example.com and insert a string before the example.com portion, type the following: `HTTP.REQ.HOSTNAME.BEFORE_STR` ("example.com")	A string expression that describes the new string you want to insert. For example, if you want to insert the new string en. before the string example in the hostname, type en followed by a period.

INSERT_AFTER: Inserts a new string after the designated string in the HTTP header or body	A string expression that describes the string after which you want to insert a new string. For example, if you want to find the hostname www.example.com, and insert a string after the www. portion, type the following: `HTTP.REQ.HOSTNAME.AFTER_STR ("www.")`	A string expression that describes the new string you want to insert. For example, if you want to insert the new string en. after the string www. in the hostname, type en followed by a period.
REPLACE: Replaces the designated string in the HTTP header or body with a different string	A string expression that describes the string you want to replace with a new string. For example, if you want to replace the entire hostname in the Host header, type `HTTP.REQ.HOSTNAME.SERVER`.	A string expression that describes the new string you want to insert. For example, if you want to replace the current host header with the string web01.example.net, type `web01.example.net`.
DELETE: Deletes the designated string from the HTTP header or body.	A string expression that describes the string you want to delete. For example, if you want to find and delete the string .en in the hostname of HTTP response headers, type the following: `HTTP.RES.HEADER("Host").SUBSTR("en.")`	
DELETE_HTTP_HEADER: Deletes the designated HTTP header, including all header contents.	The name of the HTTP header you want to delete. For example, if you want to delete the cache-control header from HTTP responses, type `HTTP.RES.HEADER ("Cache-Control").`	

CORRUPT_HTTP_HEADER: Replaces the name of the given HTTP header with a corrupted name so that it will not be recognized by the receiver.	The name of the HTTP header that you want to corrupt. If the specified header occurs more than once in a request, all the occurrences are corrupted. For example, if you want to corrupt the `Host` header in an HTTP request, you can use the following rewrite action command: `add rewrite action corrupt_header_act CORRUPT_HTTP_HEADER Host.`	
REPLACE_HTTP_RES: Replace the http response with the value specified in the target field.	A string expression that describes the string you want to replace the HTTP response with. For example, type `HTTP 200 OK` You are not authorized to view this page to replace the entire HTTP response with this warning.	
REPLACE_ALL: Will replace all occurrences of a pattern in the target text reference with the value specified in the string builder expression.	The part of either the HTTP request or response where you want to carry out the replacement.	A string expression that describes the new string you want to insert.
DELETE_ALL: Delete every occurrence of the pattern specified in the target text reference.	The part of either the HTTP request or response where you want the deletion to occur.	A string pattern after which the deletion should occur.
INSERT_AFTER_ALL: Inserts the value specified by string builder expression after each occurrence of a specified pattern in the target text reference.	The part of either the HTTP request or response where you want the insertion to occur.	A string expression that describes the new string you want to insert.
INSERT_BEFORE_ALL: Inserts the value you specify before each occurrence of the pattern you specify.	The part of either the HTTP request or response that you want to delete.	A string expression that describes the new string you want to insert.

CLIENTLESS_VPN_ENCODE: Encodes the URL you specify in clientless VPN format.	The URL you want to encode.	
CLIENTLESS_VPN_ENCODE_ALL: Encodes all of the URLs you specify in clientless VPN format.	A pattern that matches the URLs you want to encode.	
CLIENTLESS_VPN_DECODE: Decodes the URL you specify from clientless VPN format and returns it as unencoded text.	The URL you want to decode.	
CLIENTLESS_VPN_DECODE_ALL: Decodes all of the URLs you specify from clientless VPN format and returns them as unencoded text.	A pattern that matches all of the URLs you want to decode.	

To create a new rewrite action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a new rewrite action and verify the configuration:

- add rewrite action <name> <type> <target> [<stringBuilderExpr>] [(**-pattern** <expression> | **-patset** <string>)] [**-bypassSafetyCheck** (YES|NO)]
- show rewrite action <name>

Example

```
> add rewrite action insertact INSERT_HTTP_HEADER "client-IP" CLIENT.IP.SRC
Done

> show rewrite action insertact

      Name: insertact
      Operation: insert_http_header   Target:Client-IP
      Value:CLIENT.IP.SRC
      BypassSafetyCheck : NO
      Hits: 0
      Undef Hits: 0
      Action Reference Count: 0
 Done
```

To modify an existing rewrite action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify an existing rewrite action and verify the configuration:

- set rewrite action <name> [**-target** <string>] [**-stringBuilderExpr** <string>] [(**-pattern** <expression> | **-patset** <string>)] [**-bypassSafetyCheck** (YES|NO)]
- show rewrite action <name>

Example

```
> set rewrite action insertact -target "Client-IP"
 Done
> show rewrite action insertact

      Name: insertact
      Operation: insert_http_header   Target:Client-IP
      Value:CLIENT.IP.SRC
      BypassSafetyCheck : NO
      Hits: 0
      Undef Hits: 0
      Action Reference Count: 0
 Done
```

To remove a rewrite action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to remove a rewrite action :

rm rewrite action <name>

Example

```
> rm rewrite action insertact
Done
```

Parameters for configuring a rewrite action

name

A name for your new action, or the name of the existing action you want to modify or remove. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what this action is supposed to do. (Cannot be changed for an existing action.)

pattern or patset

An expression that describes the rewrite operation itself. You can use either of these options:

- pattern, followed by a PCRE-format regular expression that describes the operation.
- patset, followed by a string that references an existing patset that describes the operation.

bypassSafetyCheck

Whether to bypass the built-in safety checks when adding or modifying this action. Values: YES, NO. Default: NO. For more information, see Bypassing the Safety Check.

target

A NetScaler advanced expression that describes the text to be rewritten by the rewrite action.

stringBuilderExpr

Expression specifying new value of the rewritten HTTP packet. Maximum length of the input expression is 8191. Maximum size of string that can be used inside the expression is 1499.

To configure a rewrite action by using the configuration utility

1. In the navigation pane, expand **Rewrite**, and then click **Actions**.
2. In the details pane, do one of the following:
   - To create a new action, click **Add**.
   - To modify an existing action, select the action, and then click **Open**.
3. In the **Add Rewrite Action** or **Configure Rewrite Action** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring a rewrite action" as follows (asterisk indicates a required parameter):
   - Name*—name
   - Type*—type (When you select a type of action from the **Type** list, the names of the remaining text boxes in the **Create Rewrite Action or Configure Rewrite Action** dialog box change to indicate the kind of additional information to be entered. For all **Expression** and **String Expression** text boxes, be sure to enclose strings in double quotation marks. Alternatively, you can use the **Add Expression** dialog box, as described in the procedure that follows this one.)
     - Expression (Argument 1)
     - Expression (Argument 2) (Several types do not take a second argument, in which case this text area will be greyed out.)
   - Pattern—target (Several types have implied targets, in which case this text area is greyed out.)
   - Bypass Safety Check—bypassSafetyCheck
4. Click **Create** or **OK**. A message appears in the status bar, stating that the Action has been configured successfully.
5. Repeat steps 2 through 4 to create or modify as many rewrite actions as you wish.
6. Click **Close**.

To add an expression by using the Add Expression dialog box

1. In the **Create Rewrite Action** or **Configure Rewrite Action** dialog box, under the text area for the type argument you want to enter, click **Add**.

2. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.

   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   SYS

   The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   CLIENT

   The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the **Help** window below the **Construct Expression** window (which was blank) displays help describing the purpose and use of the term you just chose.

4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

   For more information about the PI expressions language and creating expressions for responder policies, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

   If you want to test the effect of a rewrite action when used on sample HTTP data, you can use the Rewrite Expression Evaluator.

   Note: The Rewrite Expression Evaluator is only available in the configuration utility. There is no NetScaler command line version.

To evaluate a rewrite action by using the Rewrite Action Evaluator dialog box

1. In the **Rewrite Actions** details pane, select the rewrite action that you want to evaluate, and then click **Evaluate**.

2. In the **Rewrite Expression Evaluator** dialog box, specify values for the following parameters. (An asterisk indicates a required parameter.)

   - **Rewrite Action***—If the rewrite action you want to evaluate is not already selected, select it from the drop-down list. After you select a Rewrite action, the Details section displays the details of the selected Rewrite action.

   - **New***—Select **New** to open the **Create Rewrite Action** dialog box and create a new rewrite action.

   - **Modify***—Select **Modify** to open the **Configure Rewrite Action** dialog box and modify the selected rewrite action.

   - **Flow Type***—Specifies whether to test the selected rewrite action with HTTP Request data or HTTP Response data. The default is Request. If you want to test with Response data, select Response.

   - **HTTP Request/Response Data***—Provides a space for you to provide the HTTP data that the **Rewrite Action Evaluator** will use for testing. You can paste the data directly into the window, or click **Sample** to insert some sample HTTP headers.

   - **Show end-of-line**—Specifies whether to show UNIX-style end-of-line characters (\n) at the end of each line of sample HTTP data.

   - **Sample**—Inserts sample HTTP data into the **HTTP Request/Response Data** window. You can choose either GET or POST data.

   - **Browse**—Opens a local browse window so that you can choose a file containing sample HTTP data from a local or network location.

   - **Clear**—Clears the current sample HTTP data from the **HTTP Request/Response Data** window.

3. Click **Evaluate**. The Rewrite Action Evaluator evaluates the effect of the Rewrite action on the sample data that you chose, and displays the results as modified by the selected Rewrite action in the **Results** window. Additions and deletions are highlighted as indicated in the legend in the lower left-hand corner of the dialog box.

4. Continue evaluating Rewrite actions until you have determined that all of your actions have the effect that you wanted.

   - You can modify the selected rewrite action and test the modified version by clicking **Modify** to open the **Configure Rewrite Action** dialog box, making and saving your changes, and then clicking **Evaluate** again.

   - You can evaluate a different rewrite action using the same request or response data by selecting it from the **Rewrite Action** drop-down list, and then clicking **Evaluate** again.

5. Click **Close** to close the **Rewrite Expression Evaluator** and return to the **Rewrite Actions** pane.

   To delete a rewrite action, select the rewrite action you want to delete, then click **Remove** and, when prompted, confirm your choice by clicking **OK**.

Configuring a Rewrite Policy

After you create any needed rewrite action(s), you must create at least one rewrite policy to select the requests that you want the NetScaler to rewrite.

A rewrite policy consists of a rule, which itself consists of one or more expressions, and an associated action that is performed if a request or response matches the rule. Policy rules can be based on almost any part of a request or response. If a configured rule matches a request or response, the corresponding policy is triggered and the action associated with it is carried out.

Note: You can use either the NetScaler command line or the configuration utility to create and configure rewrite policies. Users who are not thoroughly familiar with the NetScaler command line and the NetScaler Policy expression language will usually find using the configuration utility much easier.

To add a new rewrite policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a new rewrite policy and verify the configuration:

- add rewrite policy <name> <expression> <action> [<undefaction>]
- show rewrite policy <name>

Example

```
> add rewrite policy policyNew "HTTP.RES.IS_VALID" insertact NOREWRITE
 Done
> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertact
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
```

To modify an existing rewrite policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify an existing rewrite policy and verify the configuration:

- set rewrite policy <name> **-rule** <expression> **-action** <action> [<undefaction>]
- show rewrite policy <name>

Example

```
> set rewrite policy policyNew -rule "HTTP.RES.IS_VALID" -action insertaction
 Done

> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertaction
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
```

To remove a rewrite policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a rewrite policy:

rm rewrite policy <name>

Example

```
> rm rewrite policy policyNew
Done
```

Parameters for configuring a rewrite policy

name

A name for the policy, or the name of the existing policy you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of request this policy was created to match. (Cannot be changed for an existing policy.)

rule

The expression that defines the rule for this policy. The expression can be a simple expression or a complex expression that contains several expressions in structured relationship to one another. Expressions are written in the NetScaler Policy Infrastructure (PI) language. For more information about PI, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

action

The name of the rewrite action associated with the policy. You can choose either one of the built-in rewrite actions, or a rewrite action you have configured. For a complete list of built-in rewrite actions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

undefAction

The action to use if the policy generates an UNDEF event. You can select the NOREWRITE, RESET, or DROP action, or configure the NetScaler use the configured global undefined action.

Any rewrite-specific undefined action you configure will override the global undefined action.

To configure a rewrite policy by using the configuration utility

1. In the navigation pane, expand **Rewrite** and click **Policies**.
2. In the details pane, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Open**.
3. In the **Create Rewrite Policy** or **Configure Rewrite Policy** dialog box, specify values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring a rewrite policy" as follows (asterisk indicates a required parameter):
   - Name*—name
   - Action*—action
   - Undefined-Result Action—undefAction
   - Expression*—expression (You can add the expression in any of three ways. First, you can click **Add** and choose an existing expression in the **Frequently Used Expressions** drop-down list. Second, you can type the expression directly into the supplied text box. For brief help and prompts, while the cursor is in the text box, hold down the **CTRL** key and then press the **Space** bar. Third, you can use the **Add Expression** dialog box, as described in “To add an expression by using the Add Expression dialog box.”)
4. Click **Create** or **OK**. A message appears in the status bar, stating that the Policy has been configured successfully.
5. Repeat steps 2 through 4 to create or modify as many rewrite actions as you wish.
6. Click **Close**. To delete a rewrite policy, select the rewrite policy you want to delete, then click **Remove** and, when prompted, confirm your choice by clicking **OK**.

Binding a Rewrite Policy

After creating a rewrite policy, you must bind it to put it into effect. You can bind your policy to Global if you want to apply it to all traffic that passes through your NetScaler, or you can bind your policy to a specific virtual server or bind point to direct only that virtual server or bind point's incoming traffic to that policy. If an incoming request matches a rewrite policy, the action associated with that policy is carried out.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order - the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is applied first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000.

Unlike most other features in the NetScaler operating system, the rewrite feature continues to evaluate and implement policies after a request matches a policy. However, the effect of a particular action policy on a request or response will often be different depending on whether it is performed before or after another action. Priority is important to get the results you intended.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind it. If you do this, you can add additional policies at any time without having to reassign the priority of an existing policy.

When binding a rewrite policy, you also have the option of assigning a goto expression (gotoPriorityExpression) to the policy. A goto expression can be any positive integer that matches the priority assigned to a different policy that has a higher priority than the policy that contains the goto expression. If you assign a goto expression to a policy, and a request or response matches the policy, the NetScaler will immediately go to the policy whose priority matches the goto expression. It will skip over any policies with priority numbers that are lower than that of the current policy, but higher than the priority number of the goto expression, and not evaluate those policies.

For more information about binding policies on the NetScaler, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Note: Rewrite policies cannot be bound to TCP-based virtual servers.

To globally bind a rewrite policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind a rewrite policy and verify the configuration:

- bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [**-type** <type>] [**-invoke** (<labelType> <labelName>)]
- show rewrite global

Example

```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)      Global bindpoint: RES_DEFAULT
       Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
       Number of bound policies: 1

 Done
```

To bind rewrite policy to a specific virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind rewrite policy to a specific virtual server and verify the configuration:

- bind lb vserver <name>@ (<serviceName>@ [**-weight** <positive_integer>]) | <serviceGroupName>@ | (**-policyName** <string>@ [**-priority** <positive_integer>] [**-gotoPriorityExpression** <expression>] [**-type** (REQUEST | RESPONSE)] [**-invoke** (<labelType> <labelName>)])
- show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
      lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
      State: DOWN
      Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
      Time since last state change: 28 days, 01:57:26.350
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
```

```
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:50
2)      Policy : cf-pol Priority:1      Inherited
 Done
```

Parameters for binding a rewrite policy

name

If you are binding this rewrite policy to a specific virtual server, the name of that virtual server.

policyName

The name of the rewrite policy you want to bind. This is a required argument.

priority

The priority assigned to this rewrite policy. The priority determines the order in which policies are evaluated, allowing the NetScaler to evaluate the most specific policy first, and more general policies in descending order, finishing with the most general policy. This is a required argument.

gotoPriorityExpression

The priority of the next policy that should be evaluated if this policy matches. If set to END, this parameter halts the policy evaluation process after evaluation of the current policy. If you are careful to assign your policy priorities in the right order, you can use this parameter to skip over policies in the event that the current policy matches, and go directly to a specific policy.

type

Bindpoint, specifying where to bind the policy.

invoke

Type of policy label invocation.

labelType

Type of policy label invocation.

labelName

Name of the label to invoke if the current policy rule evaluates to TRUE.

weight

Weight for this service. This weight is used when the system performs load balancing, giving greater priority to a specific service. It is useful when the services bound to a virtual server are of different capacity.

To globally bind a rewrite policy by using the configuration utility

1. In the navigation pane, expand **Rewrite** and click **Policies**.
2. In the details pane, select the rewrite policy you want to globally bind, and then click **Policy Manager**.
3. In the **Rewrite Policy Manager** dialog box, in the **Bind Points** menu, select **Default Global**. The **Default Global** tab appears, with a list of all rewrite policies that are currently bound to Global.
4. Click **Insert Policy** to insert a new row and display a drop-down list with all available, unbound rewrite policies.
5. Click the policy you want to bind to Global. That policy is inserted into the list of globally bound rewrite policies.
6. In the Priority column, modify the priority to any positive integer you want. For more information about this parameter, see `priority` in the table above.
7. If you want to skip over policies and go directly to a specific policy in the event that the current policy matched, in the **Goto Expression** column, modify the value to any positive integer you want. For more information about this parameter, see `gotoPriorityExzpr` in the table above.
8. Click **Apply Changes**.
9. Click **Close**. A message appears in the status bar, stating that the Policy has been configured successfully.

To bind a rewrite policy to a specific virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane list of virtual servers, select the virtual server to which you want to bind the rewrite policy, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, select the **Policies** tab. All policies configured on your NetScaler appear on the list.
4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click **OK**. A message appears in the status bar, stating that the Policy has been configured successfully.

Configuring Rewrite Policy Labels

If you want to build a more complex policy structure than is supported by single policies, you can create policy labels and then bind them as you would policies. A policy label is a user-defined point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority you configured. A policy label can include one or multiple policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and return of control to the policy that invoked the policy label.

A rewrite policy label consists of a name, a transform name that describes the type of policy included in the policy label, and a list of policies bound to the policy label. Each policy that is bound to the policy label contains all of the elements described in Configuring a Rewrite Policy.

Note: You can use either the NetScaler command line or the configuration utility to create and configure rewrite policy labels. Users who are not thoroughly familiar with the NetScaler command line and the NetScaler Policy Infrastructure (PI) language will usually find using the configuration utility much easier.

To configure a rewrite policy label by using the NetScaler command line

To add a new rewrite policy label, at the NetScaler command prompt, type the following command:

```
add rewrite policylabel <labelName> <transform>
```

For example, to add a rewrite policy label named `polLabelHTTPResponses` to group all policies that work on HTTP responses, you would type the following:

```
add rewrite policylabel polLabelHTTPResponses http_res
```

To modify an existing rewrite policy label, at the NetScaler command prompt, type the following command:

```
set rewrite policylabel <labelName> <transform>
```

Note: The set rewrite policy command takes the same options as the add rewrite policy command.

To remove a rewrite policy label, at the NetScaler command prompt, type the following command:

```
rm rewrite policylabel <name>
```

For example, to remove a rewrite policy label named `polLabelHTTPResponses`, you would type the following:

```
rm rewrite policy polLabelHTTPResponses
```

Parameters for Rewrite Policies

name

A name for the policy label, or the name of the existing policy label that you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of policy this policy label was created to contain. (Cannot be changed for an existing policy label.)

transform

The type of policy that this policy label contains. Your choices are:

- **http_req**. Groups rewrite policies that process HTTP requests.
- **http_res**. Groups rewrite policies that process HTTP responses.
- **URL**. Groups rewrite policies that process HTTP URLs.
- **text**. Groups rewrite policies that process text.
- **clientless_vpn_req**. Groups rewrite policies that process clientless VPN requests.
- **clientless_vpn_res**. Groups rewrite policies that process clientless VPN responses.

The default is http_req.

To configure a rewrite policy label by using the configuration utility

1. In the navigation pane, expand **Rewrite** and click **Policy Labels**.
2. In the details pane, do one of the following:
   - To create a new policy label, click **Add**.
   - To modify an existing policy label, select the policy, and then click **Open**.
3. In the **Create Rewrite Policy** or **Configure Rewrite Policy** dialog box, specify values for the following parameters. (An asterisk indicates a required parameter. For a term in parentheses, see the corresponding parameter in the table above.)
   - **Name*** (name)
   - **Transform*** (transform)
4. Add or remove policies from the list that is bound to the policy label.
   - To add a policy to the list, click **Insert Policy**, and choose a policy from the drop-down list. You can create a new policy and add it to the list by choosing New Policy in the list, and following the instructions in Configuring a Rewrite Policy.
   - To remove a policy from the list, select that policy, and then click **Unbind Policy**.
5. Modify the priority of each policy by editing the number in the Priority column.

   You can also automatically renumber policies by clicking **Regenerate Priorities**.

6. Click **Create** or **OK**, and then click **Close**.

   To remove a policy label, select it, and then click **Remove**. To rename a policy label, select it and then click **Rename**. Edit the name of the policy, and then click **OK** to save your changes.

Configuring the Default Rewrite Action

An undefined event is triggered when the NetScaler cannot evaluate a policy, usually because it detects a logical or other error in the policy or an error condition on the NetScaler. When the rewrite policy evaluation results in an error, the specified undefined action is carried out. Undefined actions configured at the rewrite policy level are carried out before a globally configured undefined action.

The NetScaler supports following three types of undefined actions:

undefAction NOREWRITE

Aborts rewrite processing, but does not alter the packet flow. This means that the NetScaler continues to process requests and responses that do not match any rewrite policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers, and is the default setting.

undefAction RESET

Resets the client connection. This means that the NetScaler tells the client that it must re-establish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

undefAction DROP

Silently drops the request without responding to the client in any way. This means that the NetScaler simply discards the connection without responding to the client. This action is appropriate for requests that appear to be part of a DDoS attack or another sustained attack on your servers.

Note: Undefined events can be triggered for both request and response flow specific policies.

To configure the default action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure the default action and verify the configuration:

- set rewrite param **-undefAction** (NOREWRITE | RESET | DROP)
- show rewrite param

Example

```
> set rewrite param -undefAction NOREWRITE
 Done
> show rewrite param
       Action Name: NOREWRITE
 Done
```

To configure the default action by using the configuration utility

1. In the navigation pane, expand **Protection Features**, then click **Rewrite**.
2. In the details pane, under **Rewrite Overview**, click the **Change Rewrite Settings** link. The **Set Rewrite Params** dialog box appears.
3. Under **Global Undefined-Result Action**, select an option as follows:
   - NoRewrite—NOREWRITE
   - Reset—RESET
   - Drop—DROP
4. Click **OK**. The global undefined action is set to the value you chose.

Bypassing the Safety Check

When you create a rewrite action, the NetScaler verifies that the expression you used to create the action is safe. Expressions created by the NetScaler from run-time data, such as URLs contained in HTTP requests, can cause unexpected errors. The NetScaler reports expressions that cause such errors as unsafe expressions.

In some cases, the expressions may be safe. For example, the NetScaler cannot validate an expression that contains a URL that does not resolve, even if the URL does not resolve because the Web server is temporarily unavailable.You can manually bypass the Safety Check to allow these expressions.

To bypass the safety check by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bypass the safety check and verify the configuration:

- set rewrite action <action_name> **-bypassSafetyCheck** YES
- show rewrite action <name>

Example

```
> set rewrite action insertact -bypassSafetyCheck YES
 Done
> show rewrite action insertact

      Name: insertact
      Operation: insert_http_header   Target:Client-IP
      Value:CLIENT.IP.SRC
      BypassSafetyCheck : YES
      Hits: 0
      Undef Hits: 0
      Action Reference Count: 2
 Done
```

To bypass safety check by using the configuration utility

1. In the navigation pane, expand **Protection Features**, expand **Rewrite**, and then click **Actions**.
2. In the details pane, select the rewrite action to be exempted from the safety check, and then click **Open**.
3. In the **Configure Rewrite Action** dialog box, select the **Bypass Safety Check** check box.
4. Click **OK**.

Rewrite Action and Policy Examples

The examples in this section demonstrate how to configure rewrite to perform various useful tasks. The examples occur in the server room of Example Manufacturing Inc., a mid-sized manufacturing company that uses its Web site to manage a considerable portion of its sales, deliveries, and customer support.

Example Manufacturing has two domains: example.com for its Web site and email to customers, and example.net for its intranet. Customers use the Example Web site to place orders, request quotes, research products, and contact customer service and technical support.

As an important part of Example's revenue stream, the Web site must respond quickly and keep customer data confidential. Example therefore has several Web servers and uses Citrix NetScaler appliances to balance the Web site load and manage traffic to and from its Web servers.

The Example system administrators use the rewrite features to perform the following tasks:

Example 1: Delete old X-Forwarded-For and Client-IP Headers.

Example Inc. removes old X-Forwarded-For and Client-IP HTTP headers from incoming requests.

Example 2: Adding a Local Client-IP Header.

Example Inc. adds a new, local Client-IP header to incoming requests.

Example 3: Tagging Secure and Insecure Connections.

Example Inc. tags incoming requests with a header that indicates whether the connection is a secure connection.

Example 4: Mask the HTTP Server Type.

Example Inc. modifies the HTTP Server: header so that unauthorized users and malicious code cannot use that header to determine the HTTP server software it uses.

Example 5: Redirect an External URL to an Internal URL.

Example Inc. hides information about the actual names of its Web servers and the configuration of its server room from users, to make URLs on its Web site shorter and easier to remember, and to improve security on its site.

Example 6: Migrating Apache Rewrite Module Rules.

Example Inc. moved its Apache rewrite rules to a NetScaler appliance, translating the Apache PERL-based script syntax to the NetScaler rewrite rule syntax.

Example 7: Marketing Keyword Redirection.

The marketing department at Example Inc. sets up simplified URLs for certain predefined keyword searches on the company's Web site.

Example 8: Redirect Queries to the Queried Server.

Example Inc. redirects certain query requests to the appropriate server.

Example 9: Home Page Redirection.

Example Inc. recently acquired a smaller competitor, and it now redirects requests for the acquired company's home page to a page on its own Web site.

Each of these tasks requires that the system administrators create rewrite actions and policies and bind them to a valid bind point on the NetScaler.

Example 1: Delete Old X-Forwarded-For and Client-IP Headers

Example Inc. wants to remove old X-Forwarded-For and Client-IP HTTP headers from incoming requests, so that the only X-Forwarded-For headers that appear are the ones added by the local server. This configuration can be done through the NetScaler command line or the configuration utility. The Example Inc. system administrator is an old-school networking engineer and prefers to use a CLI where possible, but wants to be sure he understands the configuration utility interface so that he can show new system administrators on the team how to use it.

The examples below demonstrate how to perform each configuration with both the CLI and the configuration utility. The procedures are abbreviated on the assumption that users will already know the basics of creating rewrite actions, creating rewrite policies, and binding policies.

- For more detailed information about creating rewrite actions, see Configuring a Rewrite Action.
- For more detailed information about creating rewrite policies, see Configuring a Rewrite Policy.
- For more detailed information about binding rewrite policies, see Binding a Rewrite Policy.

To delete old X-Forwarded and Client-IP headers from a request by using the NetScaler command line

At the NetScaler command prompt, type the following commands in the order shown:

```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER("client-ip").EXISTS' act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
```

To delete old X-Forwarded and Client-IP headers from a request by using the configuration utility

In the **Create Rewrite Action** dialog box, create two rewrite actions with the following descriptions.

Name	Type	Argument(s)
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

In the **Create Rewrite Policy** dialog box, create two rewrite policies with the following descriptions.

Name	Expression	Action
pol_check_xfor	'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'	act_del_xfor
pol_check_cip	'HTTP.REQ.HEADER("client-ip").EXISTS'	act_del_cip

Bind both policies to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_check_xfor	100	200
pol_check_xfor	200	300

All old X-Forwarded-For and Client-IP HTTP headers are now deleted from incoming requests.

Example 2: Adding a Local Client-IP Header

Example Inc. wants to add a local Client-IP HTTP header to incoming requests. This example contains two slightly different versions of the same basic task.

To add a local Client-IP header by using the NetScaler command line

At the NetScaler command prompt, type the following commands in the order shown:

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-
bind rewrite global pol_ins_client 300 END
```

To add a local Client-IP header by using the configuration utility

In the **Create Rewrite Action** dialog box, create a rewrite action with the following description.

Name	Type	Argument(s)
act_ins_client	insert_http_header	NS-Client 'CLIENT.IP.SRC'

In the **Create Rewrite Policy** dialog box, create a rewrite policy with the following description.

Name	Expression	Action
pol_ins_client	'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS'	act_ins_client

Bind both policies to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_check_xfor	100	200
pol_check_xfor	200	300

A local Client-IP HTTP header is now added to incoming requests. You can also modify the configuration above to append all IPs from X-Forwarded-For headers to the new Client-IP header, as shown below.

Example 3: Tagging Secure and Insecure Connections

Example Inc. wants to tag incoming requests with a header that indicates whether or not the connection is a secure connection. This helps the server keep track of secure connections after the NetScaler has decrypted the connections.

To implement this configuration, you would begin by creating rewrite actions with the values shown in the following tables. These actions label connections to port 80 as insecure connections, and connections to port 443 as secure connections.

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	YES

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	NO

You would then create a rewrite policy with the values shown in the following tables. These policies check incoming requests to determine which requests are directed to port 80 and which are directed to port 443. The policies then add the correct SSL header.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

Finally, you would bind the rewrite policies to NetScaler, assigning the first policy a priority of 200, and the second a priority of 300, and setting the goto expression of both policies to END.

Each incoming connection to port 80 now has an SSL:NO HTTP header added to it and each incoming connection to port 443 has an SSL:YES HTTP header added to it.

Example 4: Mask the HTTP Server Type

Example Inc. wants to modify the HTTP Server: header so that unauthorized users and malicious code cannot use the header to identify the software that the HTTP server uses.

To modify the HTTP Server: header, you would create a rewrite action and a rewrite policy with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Server_Mask	REPLACE	HTTP.RES.HEADER("Server")	"Web Server 1.0"

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

You would then globally bind the rewrite policy, assigning a priority of 100 and setting the Goto Priority Expression of the policy to END.

The HTTP Server: header is now modified to read “Web Server 1.0,” masking the actual HTTP server software used by the Example Inc. Web site.

Example 5: Redirect an External URL to an Internal URL

Example Inc. wants to hide its actual server room configuration from users to improve security on its Web servers.

To do this, you would create a rewrite action with the values as shown in the following tables. For request headers, the action in the table modifies `www.example.com` to web.hq.example.net. For response headers, the action does the opposite, translating `web.hq.example.net to www.example.com`.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Request_Server_Replace	REPLACE	HTTP.REQ.HOSTNAME.SERVER	"Web.hq.example.net"
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER("Server")	"www.example.com"

Next, you would create rewrite policies using the values shown in the following tables. The first policy checks incoming requests to see if they are valid, and if they are, it performs the Action-Rewrite-Request_Server_Replace action. The second policy checks responses to see if they originate at the server web.hq.example.net. If they do, it performs the Action-Rewrite-Response_Server_Replace action.

ne	Action Name	Undefined Action	Expression
vrite-Request_Server_Replace	Action-Rewrite-Request_Server_Replace	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ("w
vrite-Response_Server_Replace	Action-Rewrite-Response_Server_Replace	NOREWRITE	HTTP.RES.HEADER("Server").EQ("web

Finally, you would bind the rewrite policies, assigning each a priority of 500 because they are in different policy banks and therefore will not conflict. You should set the goto expression to NEXT for both bindings.

All instances of `www.example.com` in the request headers are now changed to `web.hq.example.net`, and all instances of web.hq.example.net in response headers are now changed to `www.example.com`.

Example 6: Migrating Apache Rewrite Module Rules

Example Inc., is currently using the Apache rewrite module to process search requests sent to its Web servers and redirect those requests to the appropriate server on the basis of information in the request URL. Example Inc. wants to simplify its setup by migrating these rules onto the NetScaler platform.

Several Apache rewrite rules that Example currently uses are shown below. These rules redirect search requests to a special results page if they do not have a SiteID string or if they have a SiteID string equal to zero (0), or to the standard results page if these conditions do not apply.

The following are the current Apache rewrite rules:

- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results2.html [P,L]
- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results.html [P,L]

To implement these Apache rewrite rules on the NetScaler, you would create rewrite actions with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Display_Results_NulSiteID	REPLACE	HTTP.REQ.URL	"/results2.html"
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	"/results2.html"

You would then create rewrite policies with the values as shown in the tables below.

	Action Name	Undefined Action	Expression

ts_NulSiteID	Action-Rewrite-Display_Results_NulSiteID	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MODE(IGNORECASE).EQ (!HTTP.REQ.URL.QUERY.CONTAINS("SiteId=") || HTTP.R HTTP.REQ.URL.QUERY.SET_TEXT_MODE(IGNORECASE).C
ts	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MODE(IGNORECASE).EQ HTTP.REQ.URL.QUERY.SET_TEXT_MODE(IGNORECASE).C

Finally, you would bind the rewrite policies, assigning the first a priority of 600 and the second a priority of 700, and then set the goto expression to NEXT for both bindings.

The NetScaler now handles these search requests exactly as the Web server did before the Apache rewrite module rules were migrated.

Example 7: Marketing Keyword Redirection

The marketing department at Example Inc. wants to set up simplified URLs for certain predefined keyword searches on the company's Web site. For these keywords, it wants to redefine the URL as shown below.

- External URL: `http://www.example.com/<marketingkeyword>`
- Internal URL: `http://www.example.com/go/kwsearch.asp?keyword=<marketingkeyword>`

To set up redirection for marketing keywords, you would create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target location	String expression for replacement text
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GET(1)	""go/kwsearch.aspkeyword="\"

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.c

Finally, you would bind the rewrite policy, assigning it a priority of 800. Unlike the previous rewrite policies, this policy should be the last to be applied to a request that matches its criteria. For this reason, NetScaler administrator sets its Goto Priority Expression to END.

Any request using a marketing keyword is redirected to the keyword search CGI page, whereupon a search is performed and all remaining policies are skipped.

Example 8: Redirect Queries to the Queried Server

Example Inc. wants to redirect query requests to the appropriate server, as shown here.

- Request: `GET /query.cgi?server=5HOST: www.example.com`
- Redirect URL: `http://web-5.example.com/`

To implement this redirection, you would first create a rewrite action with the values in the following table.

on Name	Type of Rewrite Action	Expression to choose target reference	String expression for re
on-Rewrite-Replace_Hostheader	REPLACE	HTTP.REQ.HEADER("Host").BEFORE_STR(".example.com")	"server-" + HTTP.REQ.URL.QUERY.

You would then create a rewrite policy with the values in the following table.

icy Name	Action Name	Undefined Action	Expression
icy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	HTTP.REQ.HEADER("Host").EQ("www.exa

Finally, you would bind the rewrite policy, assigning it a priority of 900. Because this policy should be the last policy applied to a request that matches its criteria, you set the goto expression to END.

Incoming requests to any URL that begins with `http://www.example.com/query.cgi?server=` are redirected to the server number in the query.

Example 9: Home Page Redirection

New Company, Inc. recently acquired a smaller competitor, Purchased Company, and wants to redirect the home page for Purchased Company to a new page on its own Web site, as shown here.

- Old URL: `http://www.purchasedcompany.com/*`
- New URL: `http://www.newcompany.com/products/page.htm`

To redirect requests to the Purchased Company home page, you would create rewrite actions with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_AND_QUERY	"/products/page.htm"
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	"www.newcompany.com"

You would then create rewrite policies with the values in the following table.

cy Name	Action Name	Undefined Action	Expression
cy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ("www.purchasedco
cy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ("www.purchasedcon
cy-Rewrite-Replace-URL	Action-Rewrite-Replace_URL	NOREWRITE	HTTP.REQ.IS_VALID

Finally, you would bind the rewrite policies globally, assigning the first a priority of 100, the second a priority of 200, and the third a priority of 300. These policies should be the last policies applied to a request that matches the criteria. For this reason, set the goto expression to END for the first and third policies, and to 300 for the second policy. This ensures that all remaining requests are processed correctly.

Requests to the acquired company's old Web site are now redirected to the correct page on the New Company home page.

URL Transformation

The URL transformation feature provides a method for modifying all URLs in designated requests from an external version seen by outside users to an internal URL seen only by your Web servers and IT staff. You can redirect user requests seamlessly, without exposing your network structure to users. You can also modify complex internal URLs that users may find difficult to remember into simpler, more easily remembered external URLs.

Note: Before you can use the URL transformation feature, you must enable the Rewrite feature. To enable the Rewrite feature, see Enabling the Rewrite Feature.

To begin configuring URL transformation, you create profiles, each describing a specific transformation. Within each profile, you create one or more actions that describe the transformation in detail. Next, you create policies, each of which identifies a type of HTTP request to transform, and you associate each policy with an appropriate profile. Finally, you globally bind each policy to put it into effect.

Configuring URL Transformation Profiles

A profile that describes a specific URL transformation as a series of actions. The profile functions primarily as a container for the actions, determining the order in which the actions are performed. Most transformations transform an external hostname and optional path into a different, internal hostname and path. Most useful transformations are simple and require only a single action, but you can use multiple actions to perform complex transformations.

You cannot create actions and then add them to a profile. You must create the profile first, and then add actions to it. In the CLI, creating an action and configuring the action are separate steps. Creating a profile and configuring the profile are separate steps in both the CLI and the configuration utility.

To create a URL transformation profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands, in the order shown, to create a URL transformation profile and verify the configuration. You can then repeat the second and third commands to configure additional actions:

- add transform profile <profileName> **-type** URL [**-onlyTransformAbsURLinBody** (ON|OFF)] [**-comment** <comment>]
- add transform action <name> <profileName> <priority>
- set transform action <name> [**-priority** <priority>] [**-reqUrlFrom** <expression>] [**-reqUrlInto** <expression>] [**-resUrlFrom** <expression>] [**-resUrlInto** <expression>] [**-cookieDomainFrom** <expression>] [**-cookieDomainInto** <expression>] [**-state** (ENABLED|DISABLED)] [**-comment** "<string>"]
- show transform profile <name>

Example

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping.example.com' -reqUrlInto 'www.exar
 Done
> show transform profile shoppingcart
      Name: shoppingcart
            Type: URL       onlyTransformAbsURLinBody: OFF
      Comment:
      Actions:
```

```
1)          Priority 1000   Name: actshopping       ENABLED
 Done
```

To modify an existing URL transformation profile or action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify an existing URL transformation profile or action and verify the configuration:

Note: Use a set transform profile or set transform action command, respectively. The set transform profile command takes the same arguments as does the add transform profile command, and set transform action is the same command that was used for initial configuration.

- set transform action <name> [**-priority** <priority>] [**-reqUrlFrom** <expression>] [**-reqUrlInto** <expression>] [**-resUrlFrom** <expression>] [**-resUrlInto** <expression>] [**-cookieDomainInto** <expression>] [**-state** (ENABLED|DISABLED)] [**-comment** "<string>"]
- show transform profile <name>

Example

```
> set transform action actshopping -priority 1000 -reqUrlFrom 'searching.example.net' -reqUrlInto 'www.exar
 Done
> show transform profile shoppingcart
      Name: shoppingcart
            Type: URL       onlyTransformAbsURLinBody: OFF
      Comment:
      Actions:

1)          Priority 1000   Name: actshopping       ENABLED
 Done
```

To remove a URL transformation profile and actions by using the NetScaler command line

First remove all actions associated with that profile by typing the following command once for each action:

- rm transform action <name> After you have removed all actions associated with a profile, remove the profile as shown below.
- rm transform profile <name>

Parameters for configuring URL transformation profiles

profileName

A name for your new profile, or the name of the existing profile you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

onlyTransformAbsURLinBody

Transform only absolute URLs, not relative URLs, in HTTP body text. Possible values: YES, NO. Default: NO.

name

A name for your new action, or the name of the existing action you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

priority

The priority assigned to this URL transformation profile or action. Priorities assigned to profiles control the order in which matching URL transformation profiles are performed on a single request or response. Priorities assigned to actions control the order in which actions assigned to a single profile are performed.

reqUrlFrom

A PCRE-format regular expression that describes the request URL pattern to be transformed.

reqUrlInto

A PCRE-format regular expression that describes the transformation to be performed on the URLs in matching requests.

resUrlFrom

A PCRE-format regular expression that describes the response URL pattern to be transformed.

resUrlInto

A PCRE-format regular expression that describes the transformation to be performed on URLs in matching responses.

cookieDomainFrom

Pattern of the original domain in Set-Cookie headers.

cookieDomainInto

A PCRE-format regular expression that describes the transformation to be performed on cookies in matching requests and responses. The cookie domain to be transformed is extracted from the incoming request.

state

The state of a URL transformation action. (You can disable an action instead of removing it.) Possible values: ENABLED, DISABLED. Default: ENABLED.

comment

A text string, in quotation marks, that describes the purpose of this URL transformation action. This command is optional.

To create a URL transformation profile by using the configuration utility

1. In the navigation pane, expand **Rewrite**, expand **URL Transformation**, and then click **Profiles**.
2. In the details pane, click **Add**.
3. In the **Create URL Transformation Profile** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
   - Name*—name
   - Comment—comment
   - Only transform absolute URLs in response body—onlyTransformAbsURLinBody
4. Click **Create**, and then click **Close**. A message appears in the status bar, stating that the Profile has been configured successfully.

To configure a URL transformation profile and actions by using the configuration utility

1. In the navigation pane, expand **Rewrite**, expand **URL Transformation**, and then click **Profiles**.
2. In the details pane, select the profile you want to configure, and then click **Open**.
3. In the **Configure URL Transformation Profile** dialog box, do one of the following.
   - To create a new action, click **Add**.
   - To modify an existing action, select the action, and then click **Open**.
4. Fill in the **Create URL Transformation Action** or **Modify URL Transformation Action** dialog box by typing or selecting values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
   - Action Name*—name
   - Comments—comment
   - Priority*—priority
   - Request URL from—reqUrlFrom
   - Request URL into—reqUrlInto
   - Response URL from—resUrlFrom
   - Response URL into—resUrlInto
   - Cookie Domain from—cookieDomainFrom
   - Cookie Domain into—cookieDomainInto
   - Enabled—state
5. Save your changes.
   - If you are creating a new action, click **Create**, and then **Close**.
   - If you are modifying an existing action, click **OK**.

   A message appears in the status bar, stating that the Profile has been configured successfully.
6. Repeat step 3 through step 5 to create or modify any additional actions.
7. To delete an action, select the action, and then click **Remove**. When prompted, click **OK** to confirm the deletion.
8. Click **OK** to save your changes and close the **Modify URL Transformation Profile** dialog box.

9. To delete a profile, in the details pane select the profile, and then click **Remove**. When prompted, click **OK** to confirm the deletion.

Configuring URL Transformation Policies

After you create a URL transformation profile, you next create a URL transformation policy to select the requests and responses that the NetScaler should transform by using the profile. URL transformation considers each request and the response to it as a single unit, so URL transformation policies are evaluated only when a request is received. If a policy matches, the NetScaler transforms both the request and the response.

Note: The URL transformation and rewrite features cannot both operate on the same HTTP header during request processing. Because of this, if you want to apply a URL transformation to a request, you must make sure that none of the HTTP headers it will modify are manipulated by any rewrite action.

To configure a URL transformation policy by using the NetScaler command line

You must create a new policy. On the command line, an existing policy can only be removed. At the NetScaler command prompt, type the following commands to configure a URL transformation policy and verfiy the configuration:

- add transform policy <name> <rule> <profileName>
- show transform policy <name>

Example

```
> add transform policy polsearch  HTTP.REQ.URL.SUFFIX.EQ("Searching") prosearching
 Done
> show transform policy polsearch
1)      Name: polsearch
        Rule: HTTP.REQ.URL.SUFFIX.EQ("Searching")
        Profile: prosearching
        Priority: 0
        Hits: 0
 Done
```

To remove a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a URL transformation policy:

rm transform policy <name>

Example

```
> rm transform policy polsearch
Done
```

Parameters for configuring URL transformation policies

name

A name for your new policy, or the name of the existing policy you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

rule

A NetScaler advanced expression that defines the rule for this policy. The expression can be a simple expression, or a complex expression that contains several expressions in a structured relationship. For more information about advanced expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

profileName

The name of the profile to execute when a request or response matches this policy.

To configure a URL transformation policy by using the configuration utility

1. In the navigation pane, expand **Rewrite**, expand **URL Transformation**, and then click **Policies**.

2. In the details pane, do one of the following:

   - To create a new policy, click **Add**.

   - To modify an existing policy, select the policy, and then click **Open**.

3. In the **Create URL Transformation Policy** or **Configure URL Transformation Policy** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation policies" as follows (asterisk indicates a required parameter):

   - Name*—name (Cannot be changed for a previously configured policy.)

   - Profile*—profileName

   - Expression—rule

   If you want help with creating an expression for a new policy, you can either hold down the `Control key` and press the `space bar` while your cursor is in the **Expression** text box. To create the expression, you can type it directly as described below, or you can use the **Add Expression** dialog box as described in To add an expression by using the Add Expression dialog box.

   a. Click **Prefix**, and choose the prefix for your expression.

      Your choices are:

      - **HTTP**—The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

      - **SYS**—The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

      - **CLIENT**—The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

      - **SERVER**—The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

      - **URL**—The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.

      - **TEXT**—Any text string in the request. Choose this if you want to examine a text string in the request.

      - **TARGET**—The target of the request. Choose this if you want to examine some aspect of the request target.

After you choose a prefix, the NetScaler displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom. The choices depend on which prefix you chose.

b. Select your next term.

If you chose HTTP as your prefix, your choices are REQ, which specifies HTTP requests, and RES, which specifies HTTP responses. If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you are certain which choice you want, double-click it to insert it into the **Expression** window.

c. Type a period, and then continue selecting terms from the list boxes that appear to the right of the previous list box. You type the appropriate text strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished. For more information about creating expressions for URL Transformation policies, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

4. Click **Create** or **OK**, depending on whether you are creating a new policy or modifying an existing policy.

5. Click **Close**. A message appears in the status bar, stating that the Policy has been configured successfully.

To add an expression by using the Add Expression dialog box

1. In the **Create Responder Action** or **Configure Responder Action** dialog box, click **Add**.
2. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.

   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   SYS

   The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   CLIENT

   The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   SERVER

   The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

   URL

   The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.

   TEXT

   Any text string in the request. Choose this if you want to examine a text string in the request.

   TARGET

   The target of the request. Choose this if you want to examine some aspect of the request target.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the **Help** window below the **Construct Expression** window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a

value, until your expression is finished.

Globally Binding URL Transformation Policies

After you have configured your URL transformation policies, you globally bind them to put them into effect. After global binding, any a request or response that matches a URL transformation policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the URL transformation feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you tell the NetScaler to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you tell the NetScaler to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you globally bind your policies.

For more information about binding policies on the NetScaler, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Note: URL transformation policies cannot be bound to TCP-based virtual servers.

To globally bind a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind a URL transformation policy and verify the configuration:

- bind transform global <policyName> <priority>
- show transform global

Example

```
> bind transform global polisearching 100
 Done
> show transform global
1)      Policy Name: polisearching
```

Priority: 100

Done

Parameters for globally binding URL transformation policies

policyName

The name of the URL transformation policy you want to bind.

priority

The priority assigned to this URL transformation policy. The priority determines the order in which policies are evaluated, allowing the NetScaler to evaluate the most specific policy first, and more general policies in descending order, finishing with the most general policy.

To globally bind a URL transformation policy by using the configuration utility

1. In the navigation pane, expand **Rewrite**, then expand **URL Transformation**, and then click **Policies**.
2. In the details pane, select the URL transformation policy you want to globally bind, and then click **Global Bindings**.
3. Click **Insert Policy** to insert a new row and display a drop-down list with all available, unbound URL transformation policies.
4. Click the policy you want to bind to Global. That policy is inserted into the list of globally bound URL transformation policies.
5. Repeat steps 3 and 4 to add any additional URL transformation policies you want to globally bind.
6. Click **OK** to save your changes. A message appears in the staus bar, stating that the Policy has been configured successfully.

SSL Offload and Acceleration

A Citrix® NetScaler® appliance configured for SSL acceleration transparently accelerates SSL transactions by offloading SSL processing from the server. To configure SSL offloading, you configure a virtual server to intercept and process SSL transactions, and send the decrypted traffic to the server (unless you configure end-to-end encryption, in which case the traffic is re-encrypted). Upon receiving the response from the server, the appliance completes the secure transaction with the client. From the client's perspective, the transaction seems to be directly with the server. A NetScaler configured for SSL acceleration also performs other configured functions, such as load balancing.

Configuring SSL offloading requires an SSL certificate and key pair, which you must obtain if you do not already have an SSL certificate. Other SSL-related tasks that you might need to perform include managing certificates, managing certificate revocation lists, configuring client authentication, and managing SSL actions and policies.

Note: FIPS-related options for some of the SSL configuration procedures described in this document are specific to a FIPS-enabled NetScaler. For more information about configuring a FIPS-enabled NetScaler, see FIPS.

Configuring SSL Offloading

To configure SSL offloading, you must enable SSL processing on the NetScaler appliance and configure an SSL based virtual server that will intercept SSL traffic, decrypt the traffic, and forward it to a service that is bound to the virtual server. To enable SSL offloading, you must import a valid certificate and key and bind the pair to the virtual server.

Configuring SSL Offloading

To configure SSL offloading, you must enable SSL processing on the NetScaler appliance and configure an SSL based virtual server that will intercept SSL traffic, decrypt the traffic, and forward it to a service that is bound to the virtual server. To enable SSL offloading, you must import a valid certificate and key and bind the pair to the virtual server.

Enabling SSL Processing

To process SSL traffic, you must enable SSL processing. You can configure SSL based entities, such as virtual servers and services, without enabling SSL processing, but they will not work until SSL processing is enabled.

To enable SSL processing by using the NetScaler command line

At the NetScaler command prompt, type:

- enable ns feature ssl
- show ns feature

Example

```
> enable ns feature SSL
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
1)    Web Logging              WL               OFF
2)    Surge Protection         SP               ON
3)    Load Balancing           LB               ON
.
.
.
9)    SSL Offloading           SSL              ON
.
.
.
24)   NetScaler Push           push             OFF
Done
```

To enable SSL processing by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. Under **Modes and Features**, click **Change basic features**.
3. Select the **SSL Offloading** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**. A message appears in the status bar, stating that the feature has been enabled.

Configuring Services

On the NetScaler appliance, a service represents a physical server or an application on a physical server. Once configured, services are in the disabled state until the appliance can reach the physical server on the network and monitor its status.

For details about the types of services that can be configured on the NetScaler appliance, see Load Balancing.

To add a service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a service and verify the configuration:

- add service <name> (<IP> | <serverName>) <serviceType> <port>
- show service <serviceName>

Example

```
> add service ssl1 10.102.29.252 HTTP 80
 Done
> show service ssl1
        ssl1 (10.102.29.252:80) - HTTP
        State: UP
        Last state change was at Thu Nov 12 05:26:31 2009
        Time since last state change: 0 days, 00:00:06.750
        Server Name: 10.102.29.252
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED

1)      Monitor Name: tcp-default
              State: UP       Weight: 1
              Probes: 2       Failed [Total: 0 Current: 0]
```

```
        Last response: Success - TCP syn+ack received.
        Response Time: N/A
Done
```

To modify or remove a service by using the NetScaler command line

To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service. To remove a service, use the rm service command, which accepts only the <name> argument.

Parameters for adding a service

name (Service Name)

The name of the service you are configuring. The name can begin with a letter, a number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the type of service being added. (Cannot be changed after the action has been created.)

IP (Server)

The physical IP address of the server that the service you are configuring represents. Make sure that the server is reachable by the NetScaler.

serverName

The name of the server that the service you are configuring represents.

serviceType (Protocol)

The type of data handled by the server or application that the service you are configuring represents. For example, for web traffic, add a service of type HTTP.

port (Port)

The port number on which the service sends and receives data to and from the server.

To configure a service by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Services**.
2. In the **Details** pane, do one of the following:
   - To create a new service, click **Add**.
   - To modify an existing service, select the service, and then click **Open**.
3. In the **Create Service** or **Configure Service** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for adding a service” as shown:
   - Service Name*
   - Server*
   - Protocol*
   - Port*

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. In the **Services** pane, select the service that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring an SSL-Based Virtual Server

Secure sessions require establishing a connection between the client and an SSL-based virtual server on the NetScaler appliance. The SSL virtual server intercepts SSL traffic, decrypts it and processes it before sending it to services that are bound to the virtual server.

Note: The SSL virtual server is marked as down on the NetScaler appliance until a valid certificate / key pair and at least one service are bound to it. An SSL based virtual server is a load balancing virtual server of protocol type SSL or SSL_TCP. The load balancing feature must be enabled on the NetScaler.

To add an SSL-based virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

- add lb vserver <name> (**serviceType**) <IPAddress> <port>
- show lb vserver <name>

Example

```
> add lb vserver vssl  SSL 10.102.29.133 443
 Done
> show ssl vserver vssl

        Advanced SSL configuration for VServer vssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED         Refresh Count: 0
        Session Reuse: ENABLED         Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
```

To modify or remove an SSL-based virtual server by using the NetScaler command line

To modify the load balancing properties of an SSL virtual server, use the set lb vserver command, which is just like using the add lb vserver command, except that you enter the name of an existing vserver. To modify the SSL properties of an SSL-based virtual server, use the set ssl vserver command. For more information, see Customizing the SSL Configuration.

To remove an SSL virtual server, use the rm lb vserver command, which accepts only the <name> argument.

Parameters for adding an SSL-based virtual server

name (Name)

The name of the SSL based virtual server you are configuring. The name can begin with a letter, a number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the server being added.

IPAddress (IP Address)

The IP address of the virtual server that you are adding.

serviceType (Protocol)

The service that you are adding can either be of type SSL to handle secure HTTP traffic or SSL_TCP to handle secure TCP traffic.

port (Port)

The port number on which the virtual server receives SSL traffic. This is usually set to 443 for all secure transactions.

To configure an SSL-based virtual server by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the **Details** pane, do one of the following:
   - To create a new virtual server, click **Add**.
   - To modify an existing virtual server, select the virtual server, and then click **Open**.
3. In the **Create Virtual Server (SSL Offload)** or **Configure Virtual Server (SSL Offload)** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for adding an SSL-based virtual server” as shown:
   - Name*
   - IP Address*
   - Protocol*
   - Port*

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**. In the **SSL Offload Virtual Servers** pane, select the virtual server that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Binding Services to the SSL-Based Virtual Server

For the NetScaler appliance to forward decrypted SSL data to servers in the network, services representing these physical servers must be bound to the virtual server that receives the SSL data.

Because the link between the NetScaler and the physical server is typically secure, data transfer between the appliance and the physical server does not have to be encrypted. However, you can provide end-to end-encryption by encrypting data transfer between the NetScaler and the server. For details, see Configuring SSL Offloading with End-to-End Encryption.

Note: The Load Balancing feature should be enabled on the NetScaler appliance before you bind services to the SSL based virtual server. For details on enabling Load Balancing on the NetScaler, see Load Balancing.

To bind a service to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind the service to the virtual server and verify the configuration:

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example

```
> bind lb vserver vssl ssl1
 Done
> show lb vserver vssl
        vssl (10.102.29.133:443) - SSL  Type: ADDRESS
        State: DOWN[Certkey not bound]
        Last state change was at Thu Nov 12 05:31:17 2009 (+485 ms)
        Time since last state change: 0 days, 00:08:52.130
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
```

```
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) ssl1 (10.102.29.252: 80) - HTTP State: UP    Weight: 1
 Done
```

To unbind a service from a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following command:

unbind lb vserver <name> <serviceName>

Example

unbind lb vserver vssl ssl1

Parameters for binding a service to a virtual server

name

The name of the SSL based virtual server to which you are binding the service.

serviceName

The name of the service being bound to the SSL based virtual server. The service must be configured on the NetScaler before it is bound to the virtual server.

To bind a service to a virtual server by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind the service, and click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, select the **Services** tab, and then select the check box in the **Active** column of the ssl service that you want to bind to the virtual server.
4. Click **OK**. A message appears in the status bar, stating that the service has been bound successfully

Adding or Updating a Certificate-Key Pair

For any SSL transaction, the server needs a valid certificate and the corresponding private and public key pair. The SSL data is encrypted with the server's public key, which is available through the server's certificate. Decryption requires the corresponding private key.

Because the NetScaler appliance offloads SSL transactions from the server, the server's certificate and private key must be present on the appliance, and the certificate must be paired with its corresponding private key. This certificate-key pair must then be bound to the virtual server that processes the SSL transactions.

Both the certificate and the key must be in local storage on the NetScaler appliance before they can be added to the NetScaler. If your certificate or key file is not on the appliance, upload it to the appliance before you create the pair.

Note: Certificates and keys are stored in the /nsconfig/ssl directory by default. If your certificates or keys are stored in any other location, you must provide the absolute path to the files on the NetScaler appliance. The NetScaler FIPS appliances do not support external keys (non-FIPS keys). On a FIPS appliance, you cannot load keys from a local storage device such as a hard disk or flash memory. The FIPS keys must be present in the Hardware Security Module (HSM) of the appliance.

To add a certificate-key pair by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a certificate-key pair and verify the configuration:

- add ssl certKey <certkeyName> -**cert** <string>[(-**key** <string> [-**password**]) | -**fipsKey** <string>] [-**inform** (DER | PEM)] [<passplain>] [-**expiryMonitor** (ENABLED | DISABLED) [-**notificationPeriod** <positive_integer>]]
- show ssl certKey [<certkeyName>]

Example

```
> add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -password ssl
 Done
Note: For FIPS appliances, replace -key with -fipskey
> show ssl certKey sslckey
      Name: sslckey          Status: Valid,   Days to expiration:8418
      Version: 3
      Serial Number: 01
```

```
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
        Validity
              Not Before: Jul 15 02:25:01 2005 GMT
              Not After : Nov 30 02:25:01 2032 GMT
        Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
        Public Key Algorithm: rsaEncryption
        Public Key size: 1024
Done
```

To update or remove a certificate-key pair by using the NetScaler command line

To modify the expiry monitor or notification period in a certificate-key pair, use the set ssl certkey command. To replace the certificate or key in a certificate-key pair, use the update ssl certkey command. The update ssl certkey command has an additional parameter for overriding the domain check. For both commands, enter the name of an existing certificate-key pair. To remove an SSL certificate-key pair, use the rm ssl certkey command, which accepts only the <certkeyName> argument.

Parameters for adding a certificate-key pair

certkeyName (Certificate-Key Pair Name)

The name of the certificate-key pair added to the NetScaler. Maximum length: 31. (Cannot be changed after the certificate has been added.)

cert (Certificate File Name)

The file name of the valid certificate. The certificate file should be present on the NetScaler appliance's hard-disk drive. The default path for the certificate file is /nsconfig/ssl/. If the certificate is stored at any other location, the absolute path to the file must be provided. However, Citrix does not recommend storing the certificate in a location other than the default, because this may result in inconsistency during synchronization in an HA setup. Maximum length: 63 characters.

key (Private Key File Name)

The file name of the private key used to create the certificate. The private-key file must be present on the NetScaler appliance's hard disk drive. The default path for the key file is /nsconfig/ssl/. If the key is stored at any other location, the absolute path to the file must be provided. If you are adding a Certificate-Authority (CA) certificate file, do not add a private key. This parameter is not applicable to an SSL FIPS appliance. Maximum length: 63 characters.

fipskey

The name of the FIPS key used to create the certificate. The FIPS key is created and stored inside the FIPS Hardware Security Module (HSM). This option is applicable only to

an SSL FIPS appliance. Maximum length: 63 characters.

password (Password)

The pass phrase that was used to encrypt the private key. This option can be used to load encrypted private-keys. Maximum length: 31 characters.

Note: Password protected private keys are supported only for the PEM format.

inform (Certificate Format)

The input format of the certificate and the private-key files.

The two formats supported by the system are:

PEM: Privacy Enhanced Mail

DER: Distinguished Encoding Rule

Possible values: DER, PEM

Default value: PEM

passplain

The pass phrase that was used to encrypt the private key. This option can be used to load encrypted private keys. Maximum length: 31 characters.

Note: Password protected private key is supported only for the PEM format. Maximum length: 31 characters.

expiryMonitor (Notify When Expires)

Issue an alert when the certificate is about to expire. Possible values: ENABLED, DISABLED

notificationPeriod (Notification Period)

Number of days before certificate expiration, at which to generate an alert that the certificate is about to expire. Minimum value: 10. Maximum value: 100.

noDomainCheck (No Domain Check)

When updating a certificate-key pair, override the check for matching domain names.

To add or update a certificate-key pair by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. In the **Details** pane, do one of the following:
   - To add a new certificate-key pair, click **Add**.
   - To update an existing certificate-key pair, click **Update**.
3. In the **Install Certificate** or **Update Certificate** dialog box, set the following parameters:
   - Certificate-Key Pair Name*
   - Certificate File Name*
   - Private Key File Name
   - Password
   - Certificate Format
   - Notify When Expires
   - Notification Period
   - No Domain Check (Available in the **Update Certificate** dialog box only)

   * A required parameter
4. Click **Install** or **OK**, and then click **Close**. In the **SSL Certificates** pane, select the certificate that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Binding the Certificate-Key Pair to the SSL-Based Virtual Server

An SSL certificate is an integral element of the SSL encryption and decryption process. The certificate is used during an SSL handshake to establish the identity of the SSL server.

The certificate being used for processing SSL transactions must be bound to the virtual server that receives the SSL data. If you have multiple virtual servers receiving SSL data, a valid certificate-key pair must be bound to each of them.

You can use a valid, existing SSL certificate that you have uploaded to the NetScaler appliance. As an alternative for testing purposes, you can create your own SSL certificate on the appliance. Intermediate certificates created by using a FIPS key on the NetScaler cannot be bound to an SSL virtual server.

For details on how to create your own certificate, see Managing Certificates.

Note: Citrix recommends that you use only valid SSL certificates that have been issued by a trusted certificate authority.

To bind an SSL certificate-key pair to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an SSL certificate-key pair to a virtual server and verify the configuration:

- bind ssl vserver <vServerName> -**certkeyName** <certificate-KeyPairName>
- show ssl vserver <vServerName>

Example

```
> bind ssl vserver vssl -certkeyName
sslckey
 Done
> show ssl vserver vssl

      Advanced SSL configuration for VServer vssl:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 0
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: DISABLED
```

```
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      CertKey Name: sslckey   Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
```

To unbind an SSL certificate-key pair from a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following command:

unbind ssl vserver <vServerName> -**certkeyName** <string>

Example

```
unbind ssl vserver vssl -certkeyName sslckey
```

Parameters for binding the certificate-key pair to the virtual server

vServerName

The name of the SSL based virtual server to which you are binding the certificate-key pair.

certkeyName

The name of the certificate-key pair that you are binding to the virtual server. This certificate-key pair should already be configured on the NetScaler.

To bind an SSL certificate-key pair to a virtual server by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server to bind the certificate key to, and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, click **SSL Settings**.
4. In the **Available** pane, select a certificate.
5. Click **Add** to add the certificate as a server certificate. To add as an SNI certificate, in the **Add** drop-down list select **As SNI**. To add as a CA certificate, in the **Add** drop-down list select **As CA**.
6. Click **OK**. The certificate pair is bound to the virtual server.

Configuring an SSL Virtual Server for Secure Hosting of Multiple Sites

Virtual hosting is used by Web servers to host more than one domain name with the same IP address. The NetScaler supports hosting of multiple secure domains by offloading SSL processing from the Web servers using transparent SSL services or vserver-based SSL offloading. However, when multiple Web sites are hosted on the same virtual server, the SSL handshake is completed before the expected host name is sent to the virtual server. As a result, the NetScaler cannot determine which certificate to present to the client after a connection is established. This problem is resolved by enabling Server Name Indication (SNI) on the virtual server. SNI is a Transport Layer Security (TLS) extension used by the client to provide the host name during handshake initiation. Based on the information provided by the client in the SNI extension, the NetScaler presents the corresponding certificate to the client.

A wildcard SSL Certificate helps enable SSL encryption on multiple subdomains if the domains are controlled by the same organization and share the same second-level domain name. For example, a wildcard certificate issued to a sports network using the common name "*.sports.net" can be used to secure domains, such as "login.sports.net" and "help.sports.net" but not "login.ftp.sports.net."

You can bind multiple server certificates to a single SSL virtual server or transparent service using the -SNICert option. These certificates are issued by the virtual server or service if SNI is enabled on the virtual server or service.You can enable SNI at any time.

To bind multiple server certificates to a single SSL virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure SNI and verify the configuration:

- set ssl vserver <vServerName>@ [**-SNIEnable** (ENABLED | DISABLED)]
- bind ssl vserver <vServerName>@ **-certkeyName** <string> **-SNICert**
- show ssl vserver <vServerName>

  To bind multiple server certificates to a transparent service by using the NetScaler command line, replace vserver with service and **vservername** with **servicename** in the above commands.

  Note: The SSL service should be created with **-clearTextPort** 80 option.

Example

```
set ssl vserver v1 -snI ENABLED
bind ssl vserver v1 -certkeyName serverabc -SNICert
sh ssl vserver v1
Advanced SSL configuration for VServer v1:
…
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: ENABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
1)CertKey Name: servercert Server Certificate
1)CertKey Name: abccert Server Certificate for SNI
2)CertKey Name: xyzcert Server Certificate for SNI
3)CertKey Name: startcert Server Certificate for SNI
1)Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

Parameters for configuring SNI

vServerName

The name of the SSL virtual server on which SNI is enabled.

serviceName

The name of the SSL transparent service on which SNI is enabled.

SNIEnable

The state of the SNI feature on virtual server.

Possible values: ENABLED, DISABLED.

Default value: DISABLED

SNICert

The name of the certificate keys.

To bind multiple server certificates to a single SSL virtual server or transparent SSL service by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers** or **Services**.
2. In the details pane, select the virtual server or service on which SNI is to be enabled, and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** or **Configure Service** dialog box, on the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, under **Others**, select the **SNI Enable** check box.
5. Click **OK**.
6. On the **SSL Settings** tab, under **Available**, select a certificate.
7. In the **Add** drop-down list select **As SNI**.
8. To add more certificates, repeat step 7.
9. Under **Configured**, verify that the certificate is added as a server certificate for SNI.
10. Click **OK**.

Managing Certificates

An SSL certificate, which is an integral part of any SSL transaction, is a digital data form (X509) that identifies a company (domain) or an individual. The certificate has a public key component that is visible to any client that wants to initiate a secure transaction with the server. The corresponding private key, which resides securely on the NetScaler appliance, is used to complete asymmetric key (or public key) encryption and decryption.

You can obtain an SSL certificate and key in one of three ways:

- From an authorized certificate authority (CA), such as VeriSign
- By using an existing SSL certificate and key
- By generating a new SSL certificate and key on the NetScaler

Caution: Citrix recommends that you use certificates and keys obtained from authorized CAs such as VeriSign for all your SSL transactions. Certificates and keys generated on the NetScaler appliance should be used for testing purposes only, not in any live deployment.

Obtaining a Certificate from a Certificate Authority

A certificate authority (CA) is an entity that issues digital certificates for use in public key cryptography. Certificates issued or signed by a CA are automatically trusted by applications, such as web browsers, that conduct SSL transactions. These applications maintain a list of the CAs that they trust. If the certificate being used for the secure transaction is signed by any of the trusted CAs, the application proceeds with the transaction.

To obtain an SSL certificate from an authorized CA, you must create a private key, use that key to create a certificate signing request (CSR), and submit the CSR to the CA. The only special characters allowed in the file names are underscore and dot.

Creating a Private Key

The private key is the most important part of a digital certificate. By definition, this key is not to be shared with anyone and should be kept securely on the NetScaler appliance. Any data encrypted with the public key can be decrypted only by using the private key.

The NetScaler supports two encryption algorithms, RSA and DSA, for creating private keys. You can submit either type of private key to the CA. The certificate that you receive from the CA is valid only with the private key that was used to create the CSR, and the key is required for adding the certificate to the NetScaler.

Caution: Be sure to limit access to your private key. Anyone who has access to your private key can decrypt your SSL data.

All SSL certificates and keys are stored in the /nsconfig/ssl folder on the NetScaler appliance. For added security, you can use the Data Encryption Standard (DES) or triple DES (3DES) algorithm to encrypt the private key stored on the NetScaler appliance.

To create an RSA private key by using the NetScaler command

At the NetScaler command prompt, type the following command:

create ssl rsakey <keyFile> <bits> [-**exponent** (3 | F4)] [-**keyform** (DER | PEM)]

Example

create ssl rsakey Key-RSA-1 1024 -exponent F4 - keyform PEM

To create a DSA private key by using the NetScaler command

At the NetScaler command prompt, type the following command:

create ssl dsakey <keyfile> <bits> [-**keyform** (DER | PEM)]

Example

create ssl dsakey Key-DSA-1 1024 -keyform PEM

Parameters for creating a private key

RSAKeyFileName (Key Filename)

The name of the RSA cipher based key file that you are creating on the NetScaler.

DSAKeyFileName (Key Filename)

The name of the DSA cipher based key file that you are creating on the NetScaler.

bits (Key Size)

The size, in bits, of the private key for the RSA or DSA key. Minimum value: 512. Maximum value: 2048.

KeyForm (Key Format)

The format in which the created key is stored on the NetScaler appliance. Possible values: PEM, DER. Default: PEM.

exponent (Public Exponent Value)

The public exponent value for the RSA key. This is part of the cipher algorithm and is required for creating the RSA key. Possible values: F4 (Hex: 0x10001), 3 (Hex: 0x3). Default: F4.

To create an RSA private key by using the configuration utility

1. In the navigation pane, click **SSL**.
2. In the **SSL** pane, under **SSL Keys**, click **Create RSA Key**.
3. In the **Create RSA Key** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a private key” as shown:
   - Key Filename*
   - Key Size (bits)*
   - Public Exponent Value
   - Key Format

   * A required parameter
4. Click **Create**, and then click **Close**.

To create an DSA private key by using the configuration utility

1. In the navigation pane, click **SSL**.
2. In the **SSL** pane, under **SSL Keys**, click **Create DSA Key**.
3. In the **Create DSA Key** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a private key” as shown:
   - Key Filename*
   - Key Size (bits)
   - Public Exponent Value
   - Key Format

   * A required parameter
4. Click **Create**, and then click **Close**.

Creating a Certificate Signing Request

The certificate signing request (CSR) is a collection of information, including the domain name, other important company details, and the private key to be used to create the certificate. To avoid generating an invalid certificate, make sure that the details you provide are accurate.

To create a certificate signing request by using the NetScaler command line

At the NetScaler command prompt, type the following command:

create ssl certreq <reqFile> **-keyFile** <input_filename> | **-fipsKeyName** <string>) [**-keyForm** (DER | PEM) {-PEMPassPhrase }] **-countryName** <string> **-stateName** <string> **-organizationName** <string> [**-organizationUnitName** <string>] [**-localityName** <string>] [**-commonNam**e <string>] [**-emailAddress** <string>] {**-challengePassword** } [**-companyName** <string>]

Example

```
> create ssl certreq csreq1 -keyfile ramp -keyform PEM -countryName IN -stateName Karnataka -localityNam
 Done
```

Parameters for creating a certificate-signing request

reqFile (Request File Name)

The file name of the certificate-signing request. The file is stored in the /nsconfig/ssl directory by default.

keyFile (Key File Name)

The private key used to create the certificate-signing request, which then becomes part of the certificate-key pair. The private key can be either an RSA or a DSA key. It must exist in the NetScaler appliance's local storage. The keys are stored by default in the /nsconfig/ssl directory.

keyform (Key Format)

The format in which the key is stored on the NetScaler appliance. The NetScaler can store keys in either the PEM or the DER format. PEM is the default storage format.

Note: Command line users are prompted for the following information. In the configuration utility, the fields for entering this information are in a different order than shown here, and the names of some of the fields are slightly different.

Country Name

The two letter ISO code for your country (for example, US for United States).

State or Province Name (State or Province Name)

The full name for the state or province where your organization is located. Do not abbreviate.

Locality Name (City)

The name of the city or town in which your organization's head office is located.

Organization Name (Organization Name)

The name of the organization that will use this certificate. The organization name (corporation, limited partnership, university, or government agency) must be registered with some authority at the national, state, or city level. Use the legal name under which the organization is registered. Do not abbreviate the organization name and do not use the following characters in the name: < > ~ ! @ # 0 ^ * / ( )?.

Organization Unit Name (Organization Unit Name)

The name of the division or section in the organization that will use the certificate.

Common Name (Common Name)

The fully qualified domain name (FQDN) for the company or web site. The common name must match the name used by DNS servers to do a DNS lookup of your server (for example, www.mywebsite.com if your URL is http://www.mywebsite.com). Most browsers use this information for authenticating the server's certificate during the SSL handshake. If the server name in the URL does not match the common name in the server certificate, the browser terminates the SSL handshake or prompts the user with a warning message.

Caution: Do not use wildcard characters, such as * or ?, and do not use an IP address as the common name. The common name should be without the protocol specifier <http://> or <https://>.

Challenge Password (Challenge Password)

The challenge password for this certificate. If the input key specified is an encrypted key, the user will be prompted to enter the PEM pass-phrase that was used to encrypt the key.

Optional Company Name (Company Name)

An additional name for the company/web site.

emailAddress (Contact Email)

The contact person's e-mail address. This will be publicly displayed as part for the certificate. Provide an e-mail address that is monitored by an administrator who can be contacted about the certificate.

To create a certificate signing request by using the configuration Utility

1. In the navigation pane, click **SSL**.
2. Under **SSL Certificates**, click **Create Certificate Request**.
3. In the **Create Certificate Request** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a certificate –signing request" as shown:
   - Request File Name*
   - Key File Name*
   - Key Format
   - PEM Passphrase (For Encrypted Key)
   - Common Name
   - Country Name*
   - State or Province Name*
   - City
   - Organization Name*
   - Organization Unit Name
   - Challenge Password
   - Company Name
   - Contact Email

   * A required parameter
4. Click **Create**, and then click **Close**. The certificate signing request you created is saved on the NetScaler in the specified location.

Submitting the CSR to the CA

Most CAs accept certificate submissions by email. The CA will return a valid certificate to the email address from which you submit the CSR.

Importing Existing Certificates and Keys

If you want to use certificates and keys that you already have on other secure servers or applications in your network, you can export them, and then import them to the NetScaler appliance. You might have to convert exported certificates and keys before you can import them to the NetScaler.

For the details of how to export certificates from secure servers or applications in your network, see the documentation of the server or application from which you want to export.

Note: For installation on the NetScaler, key and certificate names cannot contain spaces or special characters other than those supported by the UNIX file system. Follow the appropriate naming convention when you save the exported key and certificate.

A certificate and private key pair is commonly sent in the PKCS#12 format. The NetScaler supports PEM and DER formats for certificates and keys. To convert PKCS#12 to PEM or DER, or PEM or DER to PKCS#12, see Converting the Format of SSL Certificates for Import or Export.

The NetScaler appliance does not support PEM keys in PKCS#8 format. However, you can convert these keys to a supported format by using the OpenSSL interface, which you can access from the NetScaler command line or the configuration utility. Before you convert the key, you need to verify that the private key is in PKCS#8 format. Keys in PKCS#8 format typically start with the following text:

-----BEGIN ENCRYPTED PRIVATE KEY-----

leuSSZQZKgrgUQ==

-----END ENCRYPTED PRIVATE KEY-----

To open the OpenSSL interface from the NetScaler command line

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials.
3. At the NetScaler command prompt, type shell.
4. At the shell prompt type openssl.

To open the ssl interface from the NetScaler configuration utility

1. In the navigation pane, click **SSL**.
2. In the details pane, under **Tools**, click **OpenSSL interface**.

To convert a non-supported PKCS#8 key format to an encrypted supported key format by using the OpenSSL interface

At the OpenSSL prompt, type one of the following commands, depending on whether the non-supported key format is of type rsa or dsa:

- rsa- **in** <PKCS#8 Key Filename> -**des3** -**out** <encrypted Key Filename>
- dsa -**in** <PKCS#8 Key Filename> -**des3** -**out** <encrypted Key Filename>

To convert a non-supported PKCS#8 key format to an unencrypted key format by using the OpenSSL interface

At the OpenSSL prompt, type the following commands, depending on whether the non-supported key format is of type rsa or dsa:

- rsa -**in** <PKCS#8 Key Filename> -**out** <unencrypted Key Filename>
- dsa -**in** <PKCS#8 Key Filename> -**out** <unencrypted Key Filename>

Parameters for converting an unsupported key format to a supported key format

<PKCS#8 Key Filename>

The input file name of the incompatible PKCS#8 private key.

<encrypted Key Filename>

The output file name of the compatible encrypted private key in PEM format.

<unencrypted Key Filename>

The output file name of the compatible unencrypted private key in PEM format.

Generating a Self-Signed Certificate

The NetScaler appliance has a built in CA tools suite that you can use to create self-signed certificates for testing purposes.

Caution: Because these certificates are signed by the NetScaler itself, not by an actual CA, you should not use them in a production environment. If you attempt to use a self-signed certificate in a production environment, users will receive a "certificate invalid" warning each time the virtual server is accessed.

The NetScaler supports creation of the following types of certificates

- Root-CA certificates
- Intermediate-CA certificates
- End-user certificates
  - server certificates
  - client certificates

Before generating a certificate, create a private key and use that to create a certificate signing request (CSR) on the appliance. Then, instead of sending the CSR out to a CA, use the NetScaler CA Tools to generate a certificate.

For details on how to create a private key and a CSR, see Obtaining a Certificate from a Certificate Authority.

To create a certificate by using a wizard

1. In the navigation pane, click **SSL**.
2. In the details pane, under **Getting Started**, select the wizard for the type of certificate that you want to create.
3. Follow the instructions on the screen. For more information about specific parameters, see Parameters for creating a self-signed certificate.

To create a Root-CA certificate by using the NetScaler command line

At the NetScaler command prompt, type the following command:

create ssl cert <certFile> <reqFile> <certType> [-**keyFile** <input_filename>] [-**keyform** (**DER** | **PEM**)] [-**days** <positive_integer>]

Example

```
> create ssl cert certi1 csreq1 ROOT_CERT -keyFile
rsa1 -keyForm PEM -days 365
  Done
```

To create an Intermediate-CA certificate or end-user certificate by using the NetScaler command line

At the NetScaler command prompt, type the following command:

create ssl cert <certFile> <reqFile> <certType> [-**keyFile** <input_filename>] [-**keyform** (**DER** | **PEM**)] [-**days** <positive_integer>] [-**certForm** (**DER** | **PEM**)] [-**CAcert** <input_filename>] [-**CAcertForm** (**DER** | **PEM**)] [-**CAkey** <input_filename>] [-**CAkeyForm** (**DER** | **PEM**)] [-**CAserial** <output_filename>]

Example

```
> create ssl cert certsy csr1 INTM_CERT -CAcert cert1
-CAkey rsakey1 -CAserial 23
Done
```

Parameters for creating a self-signed certificate

certFile (Certificate File Name)

The name of the generated certificate file. The newly created certificate file is stored by default in the /nsconfig/ssl/ directory.

reqFile (Certificate Request File Name)

The certificate signing request (CSR) file that is used to generate the certificate.

certType (Certificate Type)

The type of the certificate being created. You can create a Root Certificate, an Intermediate Certificate, a Client Certificate or a Server Certificate. Select one of the

following options

- **ROOT_CERT:** Specifies a self-signed Root-CA certificate. If you choose this setting, you must also set the -keyFile parameter. The generated Root-CA certificate can be used for signing end-user certificates (Client/Server) or to create Intermediate-CA certificates.
- **INTM_CERT:** Specifies an Intermediate-CA certificate.
- **CLNT_CERT:** Specifies an end-user client certificate that is used for client authentication.
- **SRVR_CERT:** Specifies an SSL server certificate to be used on physical SSL servers for an SSL backend-encryption setup.

The parameters CAcert, CAkey, and CAserial, are mandatory when creating an intermediate, client, or server certificate.

keyFile (Key File Name)

The private key used to create the certificate. You can either use an existing RSA or DSA key that you own or create a new private key on the NetScaler. This file is required only when creating a self-signed Root-CA certificate. The key file is stored in the /nsconfig/ssl directory by default.

Note: If the input key specified is an encrypted key, the user will be prompted to enter the PEM pass-phrase that was used for encrypting the key.

keyform (Key Format)

The file format in which the private key is stored. Possible values: PEM, DER. Default: PEM.

days (Validity Period)

The number of days for which the created certificate will be valid. The certificate is valid from the time and day (system time) of its creation to the number of days specified in this field. Minimum value: 1. Maximum value: 3650. Default: 365 days.

certForm (Certificate Format)

The format in which to save the certificate. Possible values: PEM, DER. Default: PEM.

CAcert (CA Certificate File Name)

The CA certificate file that will issue and sign the Intermediate-CA certificate or the end-user certificates (Client/Server). The default input path for the CA certificate file is /nsconfig/ssl/.

CAcertForm (CA Certificate File Format)

The format in which to store the CA certificate. Possible values: PEM, DER. Default: PEM.

CAkey (CA Key File Name)

The private key associated with the CA certificate that is used to sign the Intermediate-CA certificate or the end-user certificates (Client/Server). If the CA key file is password protected, the user will be prompted to enter the pass-phrase used when encrypting the key.

CAkeyForm (CA Key File Format)

The file format in which the private key of the CA certificate is stored. Possible values: PEM, DER. Default: PEM.

CAserial (CA Serial Number File)

The serial number file maintained for the CA certificate. The file will contain the serial number of the next certificate to be issued/signed by the CA (-CAcert). If the specified file does not exist, a new file will be created. The NetScaler stores the newly generated file in the /nsconfig/ssl/ directory by default.

Note: Specify the proper path of the existing serial file. Otherwise, a new serial file will be created, and that can change the certificate serial numbers assigned by the CA certificate to each of the certificate it signs.

To create a Root-CA certificate by using the configuration utility

1. In the navigation pane, click **SSL**.
2. Under **SSL Certificates**, click **Create Certificate**.
3. In the **Create Certificate** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a self-signed certificate” as shown:
   - Certificate File Name*
   - Certificate Format
   - Certificate Type
   - Certificate Request File Name*
   - Key File Name*
   - Key Format
   - PEM Passphrase (For Encrypted Key)—If the key is encrypted, you are prompted to enter the password at run-time on the CLI.
   - Validity Period (Number of Days)

   * A required parameter

   Note: Instead of typing the file name, you can use the browse button to launch the NetScaler file browser and select the file.
4. Click **Create**, and then click **Close**. The Root-CA certificate you created is saved on the NetScaler.

To create an Intermediate-CA certificate or end-user certificate by using the configuration utility

1. In the navigation pane, click **SSL**.
2. Under **SSL Certificates**, click **Create Certificate**.
3. In the **Create Certificate** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a self-signed certificate” as shown:
   - Certificate File Name*
   - Certificate Format
   - Certificate Type
   - Certificate Request File Name*
   - PEM Passphrase (For Encrypted Key)—If the key is encrypted, you are prompted to enter the password at run-time on the CLI.
   - Validity Period (Number of Days)
   - CA Certificate File Name*
   - CA Certificate File Format
   - CA Key File Name*—CAkey
   - CA Key File Format
   - PEM Passphrase (For Encrypted CA Key)
   - CA Serial Number File*

   * A required parameter

   Note: Instead of typing the file name, you can use the browse button to launch the NetScaler file browser and select the file.
4. Click **Create**, and then click **Close**. The Intermediate-CA certificate you created is saved on the NetScaler.

Generating a Diffie-Hellman (DH) Key

The Diffie-Hellman (DH) key exchange is a way for two parties involved in an SSL transaction that have no prior knowledge of each other to agree upon a shared secret over an insecure channel. This secret can then be converted into cryptographic keying material for mainly symmetric key cipher algorithms that require such a key exchange.

This feature is disabled by default and should be specifically configured to support ciphers that use DH as the key exchange algorithm.

Note: Generating a 2048-bit DH key may take a long time (up to 30 minutes).

To generate a DH key by using the NetScaler command line

At the NetScaler command prompt, type the following command:

create ssl dhparam <dhFile> [<bits>] [-**gen** (2 | 5)]

Example

create ssl dhparam Key-DH-1 512 -gen 2

Parameters for creating a DH Key

dhFile (DH File Name)

The name of the DH key that is created. The DH key is stored in the /nsconfig/ssl directory on the appliance by default.

bits (DH Parameter Size)

The size in bits of the DH key being generated.

gen (DH Generator)

The random number required for generating the DH key. This is required as part of the DH key generation algorithm. Possible Values: 2, 5. Default Value: 2

To generate a DH key by using the configuration utility

1. In the navigation pane, click **SSL**.
2. Under **Tools**, click **Create Diffe-Hellman (DH) Key**.
3. In the Create DH Param dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for creating a DH Key” as shown:
   - DH File Name (with path)*
   - DH Parameter Size (Bits)*
   - DH Generator

   * A required parameter
4. Click **OK**.

Creating a Chain of Certificates

If the server certificate is issued by an intermediate CA that is not recognized by standard Web browsers as a trusted CA, the CA certificate(s) must be sent to the client with the server's own certificate. Otherwise, the browser terminates the SSL session after it fails to authenticate the server certificate.

You must create a chain of certificates to be sent to the client during the SSL handshake. This chain links the server certificate to its issuer (the intermediate CA). For this to work, the intermediate CA certificate file must already be installed on the NetScaler appliance, and one of the certificates in the chain must be trusted by the client application. For example, link Cert-Intermediate-A to Cert-Intermediate-B, where Cert-Intermediate-B is linked to Cert-Intermediate-C, which is a certificate trusted by the client application.

Note: The NetScaler supports sending a maximum of 10 certificates in the chain of certificates sent to the client (one server certificate and nine CA certificates).

To create a certificate chain by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a certificate chain and verify the configuration. (Repeat the first command for each new link in the chain.)

- link ssl certkey <certKeyName> <linkCertKeyName>
- show ssl certlink

Example

```
> link ssl certkey siteAcertkey CAcertkey
 Done

> show ssl certlink

linked certificate:
      1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
 Done
```

Parameters for creating a certificate chain

certKeyName

The name of the certificate-key pair that is linked to its issuer's certificate-key pair in the chain.

linkCertKeyName

The name of the issuer of the certificate that is being linked to.

The two certificate-key pairs are linked only if the certificate specified in the **certKeyName** parameter is issued by the Certificate-Authority specified in the **linkCertKeyName** parameter.

To create a certificate chain by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. Select the server certificate you want to link, and then click **Link**.
3. In **Link Server Certificate(s)**, select CA Certificate Name to be linked to.
4. Click **OK**. The server certificate is now linked to the intermediate certificate.

Generating a Server Test Certificate

The NetScaler allows you to create a test certificate for server authentication by using a GUI wizard in the configuration utility. A server certificate is used to authenticate and identify a server in an SSL handshake. A server certificate is generally issued by a trusted CA and is sent out by the server to a client who uses it to authenticate the server.

For issuing a server test certificate, the NetScaler operates as a CA. This certificate can be bound to an SSL virtual server for authentication in an SSL handshake with a client. This certificate is for testing purposes only. It should not be used in a production environment.

You can install the server test certificate on any virtual server that uses the SSL or the SSL_TCP protocol.

To generate a server test certificate by using the configuration utility

1. In the navigation pane, click **SSL**.
2. Under **SSL Certificates**, click **Create and install a Server Test Certificate**.
3. In the **Create and install a Server Test Certificate** dialog box, specify values for the following parameters:
   - Certificate File Name—name of the server test certificate
   - Fully Qualified Domain Name—the domain for which you want to secure the connection
   - Country—the name of the country or region
4. Click **OK**.

Modifying and Monitoring Certificates and Keys

To avoid downtime when replacing a certificate-key pair, you can update an existing certificate. If you want to replace a certificate with a certificate that was issued to a different domain, you must disable domain checks before updating the certificate.

To receive notifications about certificates due to expire, you can enable the expiry monitor.

Updating an Existing Server Certificate

When you remove or unbind a certificate from a configured SSL virtual server, or an SSL service, the virtual server or service becomes inactive until a new valid certificate is bound to it. To avoid downtime, you can use the update feature to replace a certificate-key pair that is bound to an SSL virtual server or an SSL service, without first unbinding the existing certificate.

To update an existing certificate-key pair by using the NetScaler command line

At the NetScaler command prompt, type the following commands to update an existing certificate-key pair and verify the configuration:

- update ssl certkey <certkeyName> **-cert** <string> **-key** <string>
- show ssl certKey <certkeyName>

Example

```
> update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem
      -key /nsconfig/ssl/pkey.pem
  Done

> show ssl certkey siteAcertkey
Name: siteAcertkey       Status: Valid
      Version: 3
      Serial Number: 02
      Signature Algorithm: md5WithRSAEncryption
      Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech
```

```
    Validity
        Not Before: Nov 11 14:58:18 2001 GMT
        Not After: Aug 7 14:58:18 2004 GMT
    Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security
    Public Key Algorithm: rsaEncryption
    Public Key size: 1024
Done
```

Parameters for updating an existing certificate-key pair

certkeyName

The name of the certificate key pair that you want to update with a new certificate or a new key, or both.

cert

The name of the new certificate with which you want to update the certificate key pair.

key

The name of the private with which key you want to update an existing certificate key pair.

Note: The new certificate and key should be in local storage on the NetScaler. If the files are not stored in the default /nsconfig/ssl folder, provide the absolute path to the files.

To update an existing certificate-key pair by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. Select the certificate you want to update, and then click **Update**.
3. Use the **Browse** button next to the **Certificate File name** and the **Key File name** and select the new certificate and key files respectively.
4. If the key in encrypted, in the **Password** text box, type the password used to encrypt the key.
5. Click **OK**. In **SSL Certificates** pane, select the certificate that you just updated and verify that the settings displayed at the bottom of the screen are correct.

Disabling Domain Checks

When an SSL certificate is replaced on the NetScaler, the domain name mentioned on the new certificate should match the domain name of the certificate being replaced. For example, if you have a certificate issued to abc.com, and you are updating it with a certificate issued to def.com, the certificate update fails.

However, if you want the server that has been hosting a particular domain to now host a new domain, you can disable the domain check before updating its certificate.

To disable the domain check for a certificate by using the NetScaler command line

At the NetScaler command prompt, type the following commands to disable the domain check and verify the configuration:

- update ssl certKey <certkeyName> **-noDomainCheck**
- show ssl certKey <certkeyName>

Example

```
> update ssl certKey sv -noDomainCheck
 Done
> show ssl certkey sv
    Name: sv
    Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
    Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
    Format: PEM
    Status: Valid,   Days to expiration:9349
    Certificate Expiry Monitor: DISABLED
 Done
```

To disable the domain check for a certificate by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. Select the certificate you want to update, and then click **Update**.
3. Select **No Domain Check**, and then click **OK**. The domain check for the certificate is now disabled.

Enabling the Expiry Monitor

An SSL certificate is valid for a specific period of time. A typical deployment includes multiple virtual servers that process SSL transactions, and the certificates bound to them can expire at different times. An expiry monitor configured on the NetScaler appliance creates entries in the appliance's syslog and nsaudit logs when a certificate configured on the appliance is due to expire.

If you want to create SNMP alerts for certificate expiration, you must configure them separately.

For information about monitoring on the NetScaler, see Monitors.

To enable an expiry monitor for a certificate by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable an expiry monitor for a certificate and verify the configuration:

- set ssl certKey <certkeyName> [**-expiryMonitor** (ENABLED | DISABLED) [-**notificationPeriod** <positive_integer>]]
- show ssl certKey <certkeyName>

Example

```
> set ssl certKey sv -expiryMonitor ENABLED –notificationPeriod 60
  Done

> show ssl certkey sv
Name: sv
    Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
    Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
    Format: PEM
    Status: Valid,   Days to expiration:9349
    Certificate Expiry Monitor: ENABLED
    Expiry Notification period: 60 days
  Done
```

Parameters for enabling an expiry monitor

certKeyName

The name of the certificate-key pair whose expiry monitor is configured.

expiryMonitor

Enable or disable the expiry monitor for the certificate-key pair

notificationPeriod

The number of days in advance that the NetScaler should warn about a certificate that is about to expire.

To enable an expiry monitor for a certificate by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Certificates**.
2. Select the certificate you want to update, and then click **Update**.
3. Select the **Enable** option.
4. In the **Notification Period** text box, type the required notification period value.

   Note: The notification period parameter can be set to any value between 10 and 100 days and the default notification period is 30 days.
5. Click **OK**. In the **SSL Certificates** pane, select the certificate that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Using Global Site Certificates

A global site certificate is a special-purpose server certificate whose key length is greater than 128 bits. A global site certificate consists of a server certificate and an accompanying intermediate-CA certificate. You must import the global site certificate and its key from the server to the NetScaler appliance.

How Global Site Certificates Work

Export versions of browsers use 40-bit encryption to initiate connections to SSL Web-servers. The server responds to connection requests by sending its certificate. The client and server then decide on an encryption strength based on the server certificate type:

- If the server certificate is a normal certificate and not a global site certificate, the export client and server complete the SSL handshake and use 40-bit encryption for data transfer.
- If the server certificate is a global site certificate (and if the export client feature is supported by the browser), the export client automatically upgrades to 128-bit encryption for data transfer.

If the server certificate is a global site certificate, the server sends its certificate, along with the accompanying intermediate-CA certificate. The browser first validates the intermediate-CA certificate by using one of the Root-CA certificates that are normally included in web browsers. Upon successful validation of the intermediate-CA certificate, the browser uses the intermediate-CA certificate to validate the server certificate. Once the server is successfully validated, the browser renegotiates (upgrades) the SSL connection to 128-bit encryption.

With Microsoft's Server Gated Cryptography (SGC), if the Microsoft IIS server is configured with an SGC certificate, export clients that receive the certificate renegotiate to use128-bit encryption.

Importing a Global Site Certificate

To import a global site certificate, first export the certificate and server key from the Web server. Global site certificates are generally exported in some binary format, therefore, before importing the global site certificate, convert the certificate and key to the PEM format.

Note: For more information about exporting certificates and keys for your server type, see Exporting Existing Certificates and Keys.

To import a global site certificate

1. Using a text editor, copy the server certificate and the accompanying intermediate-CA certificate into two separate files.

   The individual PEM encoded certificate will begin with the header ----- BEGIN CERTIFICATE----- and end with the trailer -----END CERTIFICATE-----.

2. Use an SFTP client to transfer the server certificate, intermediate-CA certificate, and server-key to the NetScaler.

3. Use the following OpenSSL command to identify the server certificate and intermediate-CA certificate from the two separate files.

   Note: You can launch the OpenSSL interface from the configuration utility. In the navigation pane, click **SSL**. In the details pane, under **Tools**, click **Open SSL interface**.

   > openssl x509 -in >path of the CA cert file< ▫text

   X509v3 Basic Constraints:
   CA:TRUE
   X509v3 Key Usage:
   Certificate Sign, CRL Sign
   Netscape Cert Type:
   SSL CA, S/MIME CA

   > openssl x509 -in >path of the server certificate file< -text

   X509v3 Basic Constraints:
   CA:FALSE
   Netscape Cert Type:
   SSL Server

4. At the FreeBSD shell prompt, enter the following command:

   openssl x509 -in cert.pem -text | more

   Where **cert.pem** is one of the two certificate files.

   Read the **Subject** field in the command output. For example,

   Subject: C=US, ST=Oregon, L=Portland,
   O=mycompany, Inc., OU=IT, CN=www.mycompany.com

   If the CN field in the Subject matches the domain-name of your Web site, then this is the server certificate and the other certificate is the accompanying intermediate-CA

certificate.

5. Use the server certificate and its private key) to create a certificate key pair on the NetScaler. For details on creating a certificate-key pair on the NetScaler, see Adding a Certificate Key Pair.

6. Add the intermediate-CA certificate on the NetScaler. Use the server certificate you created in step 4 to sign this intermediate certificate. For details on creating an Intermediate-CA certificate on the NetScaler, see Generating a Self-Signed Certificate.

Converting the Format of SSL Certificates for Import or Export

A NetScaler appliance supports the PEM and DER formats for SSL certificates. Other applications, such as client browsers and some external secure servers, require various public key cryptography standard (PKCS) formats. The NetScaler can convert the PKCS#12 format (the personal information exchange syntax standard) to PEM or DER format for importing a certificate to the appliance, and can convert PEM or DER to PKCS#12 for exporting a certificate. For additional security, conversion of a file for import can include encryption of the private key with the DES or DES3 algorithm.

Note: If you use the configuration utility to import a PKCS#12 certificate, and the password contains a dollar sign ($), backquote (`), or escape (\) character, the import may fail. If it does, the `ERROR: Invalid password` message appears. If you must use a special character in the password, be sure to prefix it with an escape character (\) unless all imports are performed by using the NetScaler command line.

To convert the format of a certificate by using the NetScaler command line

At the NetScaler command prompt, type the following command:

Convert ssl pkcs12 <outfile> [-**import** [-**pkcs12File** <inputFilename>] [-**des** | -**des3**] [-**export** [-**certFile** <inputFilename>] [-**keyFile** <inputFilename>]] During the operation, you are prompted to enter an import password or an export password. For an encrypted file, you are also prompted to enter a passphrase.

Example

convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.pfx -des

convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -keyFile Key-Client-1

Parameters for Converting the Format of a Certificate

outfile (Output File Name)

The name of, and optionally the path to, the output file that contains the certificate and the private key after converting from PKCS#12 to PEM format. The default output path for the file is /nsconfig/ssl/. This is a required parameter. Maximum value: 63 characters.

pkcsFile (PKCS12 File Name)

The name of, and optionally the path to, the PKCS#12 file. If the **-import** option is specified, this is the input file name that contains the certificate and the private key in PKCS#12 format. If the **-export** option is specified, this is the output file name that contains the certificate and the private key after converting from PEM to PKCS#12 format. The default input path is /nsconfig/ssl/. Maximum value: 63 characters.

import

Convert the certificate and private key from PKCS#12 format to PEM format.

des (Encoding Format)

Encrypt the private key with DES in CBC mode during the import operation.

des3 (Encoding Format)

Encrypt the private key with DES in EDE CBC mode (168 bit key) during the import operation.

export

Convert the certificate and private key from PEM format to PKCS#12 format.

certFile (Certificate File Name)

The certificate file to be converted from PEM to PKCS#12 format.

keyFile (Key File Name)

The private key file to be converted from PEM to PKCS#12 format.

Password (Import Password/Export Password)

In the CLI, you are prompted to enter the password during the import or export operation. In the configuration utility, you have to enter the password before the operation begins. The import password is the password that was entered while creating the PKCS#12 file. For the export password, you enter a new password that will be required when installing the certificate into the client browser.

PEMPassPhrase (PEM Passphrase)

In the CLI, you are prompted to enter the passphrase during the import operation if you select des or des3 for encrypting your private key or during the export operation if you select des or des3 for encrypting your PKCS#12 file. In the configuration utility, you have to enter the passphrase before the import operation if you select an encoding format, or before the export operation if you want to encrypt your PKCS#12 file.

To convert the format of a certificate by using the configuration utility

1. In the navigation pane, click **SSL**.
2. Under **Tools**, do one of the following
   - To convert a PKCS#12 certificate and key to PEM format, click **Import PKCS#12**.

     To convert a certificate and key from PEM to PKCS#12 format, click **Export PKCS#12**.
3. In the **Import PKCS12 or Export PKCS12** dialog box, set the following parameters:
   - Output File Name*
   - PKCS12 File Name*
   - Certificate File Name*
   - Key File Name*
   - Import Password*
   - Export Password*
   - Encoding Format
   - PEM Passphrase

   * A required parameter
4. Click **OK**.

Managing Certificate Revocation Lists

A certificate issued by a CA typically remains valid until its expiration date. However, in some circumstances, the CA may revoke the issued certificate before the expiration date (for example, when an owner's private key is compromised, a company's or individual's name changes, or the association between the subject and the CA changes).

A Certificate Revocation List (CRL) identifies invalid certificates by serial number and issuer.

Certificate authorities issue CRLs on a regular basis. You can configure the NetScaler appliance to use a CRL to block client requests that present invalid certificates.

If you already have a CRL file from a CA, add that to the NetScaler. You can configure refresh options. You can also configure the NetScaler to sync the CRL file automatically at a specified interval, from either a web location or an LDAP location. The NetScaler supports CRLs in either the PEM or the DER file format. Be sure to specify the file format of the CRL file being added to the NetScaler.

If you have used the NetScaler as a CA to create certificates that are used in SSL deployments, you can also create a CRL to revoke a particular certificate. This feature can be used, for example, to ensure that self-signed certificates that are created on the NetScaler are not used either in a production environment or beyond a particular date.

Note:

By default, CRLs are stored in the /var/netscaler/ssl directory on the NetScaler appliance.

Adding an Existing CRL to the NetScaler

Before you configure the CRL on the NetScaler appliance, make sure that the CRL file is stored locally on the NetScaler. In the case of an HA setup, the CRL file must be present on both NetScaler appliances, and the directory path to the file must be the same on both appliances.

To add a CRL on the NetScaler by using the command line

At the NetScaler command prompt, type the following commands to add a CRL on the NetScaler and verify the configuration:

- add ssl crl <crlName> <crlPath> [-**inform** (DER | PEM)]
- show ssl crl [<crlName>]

Example

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM
 Done
> show ssl crl crl-one
        Name: crl-one   Status: Valid,  Days to expiration: 29
        CRL Path: /var/netscaler/ssl/CRL-one
        Format: PEM     CAcert: samplecertkey
        Refresh: DISABLED
        Version: 1
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,OU=SSL Acceleration,CN=www.ns.com/ema
        Last_update:Jun 15 10:53:53 2010 GMT
        Next_update:Jul 15 10:53:53 2010 GMT

1)      Serial Number: 00
        Revocation Date:Jun 15 10:51:16 2010 GMT
 Done
```

Parameters for adding an existing CRL

crlName (CRL Name)

The name of the CRL being added on the NetScaler.

crlPath (CRL File)

The name of the CRL file being added on the NetScaler. The NetScaler looks for the CRL file in the /var/netscaler/ssl directory by default.

inform (Format)

The format in which the CRL file is stored on the NetScaler appliance. Possible Values: PEM, DER. Default: PEM.

CAcert (CA Certificate)

The corresponding CA certificate that has issued the CRL. This is the System object identifying the CA certificate that is loaded in System. Maximum Length: 31

Note: The CA certificate should be installed before loading the CRL.

To add a CRL on the NetScaler by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **CRL**.
2. In the details pane, click **Add**.
3. In the **Add CRL** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for adding an existing CRL” as shown:
   - CRL Name*
   - CRL File*
   - Format
   - CA Certificate

   * A required parameter
4. Click **Create**, and then click **Close**. In the **CRL** pane, select the CRL that you just configured and verify that the settings displayed at the bottom of the screen are correct. For information about CRL Auto Refresh, see CRL Refresh Parameters

Configuring CRL Refresh Parameters

A CRL is generated and published by a Certificate Authority periodically or, in some cases, immediately after a particular certificate is revoked. Citrix recommends that you update CRLs on the NetScaler appliance regularly, for protection against clients trying to connect with certificates that are not valid.

The NetScaler can refresh CRLs from a web location or an LDAP directory. When you specify refresh parameters and a web location or an LDAP server, the CRL does not have to be present on the local hard disk drive at the time you execute the command. The first refresh stores a copy on the local hard disk drive, in the path specified by the CRL File parameter. The default path for storing the CRL is /var/netscaler/ssl.

To configure CRL autorefresh by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure CRL auto refresh and verify the configuration the following commands to configure CRL auto refresh and verify the configuration:

- set ssl crl <crlName> [-**refresh** (ENABLED | DISABLED)] [-**CAcert** <string>] [-**url** <URL | -**server** <ip_addr|ipv6_addr>] [-**method** HTTP | (LDAP [-**baseDN** <string>] [-**bindDN** <string>] [-**scope** (Base | One)] [-**password** <string>] [-**binary** (YES | NO)])] [-**port** <port>] [-**interval** <interval>]

- show ssl crl [<crlName>]

Example

Set CRL crl1 -refresh enabled -method ldap -inform DER -CAcert ca1 -server 10.102.192.192 -port 389 -scope

set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80 -time 00:10 -url http://10.102.192.192/cr

> sh crl

1) Name: crl1 Status: Valid, Days to expiration: 355
    CRL Path: /var/netscaler/ssl/crl1
    Format: PEM CAcert: ca1
    Refresh: ENABLED Method: HTTP
    URL: http://10.102.192.192/crl/ca1.crl Port:80
    Refresh Time: 00:10
    Last Update: Successful, Date:Tue Jul 6 14:38:13 2010
Done

CRL Refresh Parameters

crlName

The name of the CRL being refreshed on the NetScaler.

refresh

Enable or disable CRL auto refresh.

CAcert

The certificate of the CA that has issued the CRL. This CA certificate must be installed on the NetScaler. The NetScaler can update CRLs only from CAs whose certicates are installed on it.

url (URL)

The URL for the web location from which the CRL should be fetched.

server (Server IP)

The IP address of the LDAP server from which the CRL should be fetched.

method (Method)

Protocol in which to obtain the CRL refresh from a web server (HTTP) or an LDAP server. Possible Values: HTTP, LDAP. Default: LDAP.

baseDN (Base DN)

The baseDN attribute used by LDAP search to query for the **certificateRevocationList** attribute.

Note: Citrix recommends using the baseDN attribute instead of the Issuer-Name from the CA certificate to search for the CRL in the LDAP server. The Issuer-Name field may not exactly match the LDAP directory structure's DN.

bindDN (Bind DN)

The bindDN attribute to be used to access the CRL object in the LDAP repository. This is required if the access to the LDAP repository is restricted, that is, anonymous access is not allowed.

scope (Scope)

The extent of the search operation on the LDAP server. If the scope specified is Base, the search is at exactly the same level as the baseDN. If the scope specified is One, the search extends to one level below the baseDN.

password (Password)

The password used to access the CRL object in the LDAP repository. This is required if the access to the LDAP repository is restricted, that is, anonymous access is not allowed.

binary (Binary)

Set the LDAP based CRL retrieval mode to binary. Possible values: YES, NO. Default: NO.

port (Port)

The port number on which the LDAP or the HTTP server should be contacted.

interval (Interval)

The interval at which the CRL refresh should be carried out. For an instantaneous CRL refresh, specify the interval as NOW. Possible values: MONTHLY, DAILY, WEEKLY, NOW, NONE.

day (Day)

The day on which CRL refresh should be carried out. The option is not available if interval is set to DAILY.

time (Time)

The exact time in 24-hour format when the CRL refresh should be carried out.

To configure CRL autorefresh using LDAP or HTTP by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **CRL**.
2. Select the configured CRL for which you want to update refresh parameters, and then click **Open**.
3. Select the **Enable CRL Auto Refresh** option.
4. In the **CRL Auto Refresh Parameters** group, specify values for the following parameters, which correspond to parameters described in “CRL Refresh Parameters” as shown:
   - Method
   - Binary
   - Scope
   - Server IP
   - Port*
   - URL
   - Base DN*
   - Bind DN
   - Password
   - Interval
   - Day(s)
   - Time

   * A required parameter

   Note: If the new CRL has been refreshed in the external repository before its actual update time as specified by the LastUpdate field of the CRL, you should immediately refresh the CRL on the NetScaler.
5. Click **Create**. In the **CRL** pane, select the CRL that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Synchronizing CRLs

The NetScaler appliance uses the most recently distributed CRL to prevent clients with revoked certificates from accessing secure resources.

If CRLs are updated often, the NetScaler needs an automated mechanism to fetch the latest CRLs from the repository. You can configure the NetScaler to update CRLs automatically at a specified refresh interval.

The NetScaler maintains an internal list of CRLs that need to be updated at regular intervals. At these specified intervals, the appliance scans the list for CRLs that need to be updated, connects to the remote LDAP server or HTTP server, retrieves the latest CRLs, and then updates the local CRL list with the new CRLs.

Note: If CRL check is set to mandatory when the CA certificate is bound to the virtual server,and the initial CRL refresh fails, all client-authentication connections with the same issuer as the CRL are rejected as REVOKED until the CRL is successfully refreshed.

You can specify the interval at which the CRL refresh should be carried out. You can also specify the exact time.

To synchronize CRL autorefresh by using the NetScaler command line

At the NetScaler command prompt, type the following command:

set ssl crl <crlName> [-**interval** <interval>] [-**day** <integer>] [-**time** <HH:MM>]

Example

```
set ssl crl CRL-1 -refresh ENABLE -interval
MONTHLY -days 10 -time 12:00
```

Parameters for synchronizing CRL refresh

interval (Interval)

The CRL refresh interval. Possible values: DAILY, WEEKLY, MONTHLY, NONE, NOW. Specify NONE to reset a previously set interval. Specify NOW for instantaneous refresh. See also day.

day (Day)

Behavior depends on the interval setting. If **-interval** is not set, **-day** specifies the CRL refresh interval as a number of days. If **-interval** is set to MONTHLY, **-day** specifies a day of the month (1-30/31/28) If **-interval** is set to WEEKLY, **-day** specifies a day of the week, 1 to 7, where 1=Sunday and 7=Saturday. If **-interval** is DAILY, **-day** cannot be used.

time (Time)

The time of the day when the CRL should be refreshed. The time is specified in 24-hour HHMM format, where HH stands for Hours and MM stands for minutes.

To synchronize CRL refresh by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **CRL**.
2. Select the configured CRL for which you want to update refresh parameters, and then click **Open**.
3. Select the **Enable CRL Auto Refresh** option.
4. In the **CRL Auto Refresh Parameters** group, specify values for the following parameters, which correspond to parameters described in “Parameters for synchronizing CRL refresh” as shown:
   - Interval
   - Day(s)
   - Time
5. Click **Create**. In the **CRL** pane, select the CRL that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Creating a CRL on the NetScaler

Since you can use the NetScaler appliance to act as a certificate authority and create self-signed certificates, you can also revoke certificates that you have created and certificates whose CA certificate you own.

The NetScaler must revoke invalid certificates before creating a CRL for those certificates. The appliance stores the serial numbers of revoked certificates in an index file and updates the file each time it revokes a certificate. The index file is automatically created the first time a certificate is revoked.

To revoke a certificate or create a CRL by using the NetScaler command line

At the NetScaler command prompt, type the following command:

create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-**revoke** <input_filename> | -**genCRL** <output_filename>)

Example

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1

Parameters for revoking a certificate or creating a CRL on the NetScaler

CAcertFile (CA Certificate File Name)

The certificate file of the CA that issued the certificate being revoked or created.

CAkeyFile (CA Key File Name)

The private key file of the CA that issued the certificate being revoked or created.

indexFile (Index File Name)

The index file that tracks each certificate that is revoked or created. If the index file specified does not exist on the NetScaler in the path provided, a new index file is created when a CRL is created.

revoke (Revoke Certificate)

Revokes the individual certificate specified. You can revoke a certificate only if the certificate and key of the CA that issued the certificate is available on the NetScaler.

genCRL (Generate CR)

Generates a CRL with the specifed name.

To revoke a certificate or create a CRL by using the configuration utility

1. In the navigation pane, click **SSL**.
2. Under **Getting Started**, click **CRL Management**.
3. In the **CRL Management** dialog box, specify values for the following parameters, which correspond to parameters described in “CRL Refresh Parameters” as shown:
   - CA Certificate File Name*
   - CA Key File Name*
   - CA Key File Password—the password used to encrypt the key file. On the CLI, you are prompted to enter this password at run time.
   - Index File Name*
   - Choose Operation-
     - Revoke Certificate
     - Generate CRL

   * A required parameter
4. Click **Create**, and then click **Close**.

Monitoring Certificate Status with OCSP

Online Certificate Status Protocol (OCSP) is an Internet protocol that is used to determine the status of a client SSL certificate. NetScaler appliances support OCSP as defined in RFC 2560. OCSP offers significant advantages over certificate revocation lists (CRLs) in terms of timely information. Up-to-date revocation status of a client certificate is especially useful in transactions involving large sums of money and high-value stock trades. It also uses fewer system and network resources. NetScaler implementation of OCSP includes request batching and response caching.

NetScaler Implementation of OCSP

OCSP validation on a NetScaler appliance begins when the appliance receives a client certificate during an SSL handshake. To validate the certificate, the NetScaler creates an OCSP request and forwards it to the OCSP responder. To do so, the NetScaler uses a locally configured URL. The transaction is in a suspended state until the NetScaler evaluates the response from the server and determines whether to allow the transaction or reject it. If the response from the server is delayed beyond the configured time and no other responders are configured, the NetScaler will allow the transaction or display an error, depending on whether the OCSP check was set to optional or mandatory, respectively.

The NetScaler supports batching of OCSP requests and caching of OCSP responses to reduce the load on the OCSP responder and provide faster responses.

OCSP Request Batching

Each time the NetScaler receives a client certificate, it sends a request to the OCSP responder. To help avoid overloading the OCSP responder, the NetScaler can query the status of more than one client certificate in the same request. For this to work efficiently, a timeout needs to be defined so that processing of a single certificate is not inordinately delayed while waiting to form a batch.

OCSP Response Caching

Caching of responses received from the OCSP responder enables faster responses to the clients and reduces the load on the OCSP responder. Upon receiving the revocation status of a client certificate from the OCSP responder, the NetScaler caches the response locally for a predefined length of time. When a client certificate is received during an SSL handshake, the NetScaler first checks its local cache for an entry for this certificate. If an entry is found that is still valid (within the cache timeout limit), it is evaluated and the client certificate is accepted or rejected. If a certificate is not found, the NetScaler sends a request to the OCSP responder and stores the response in its local cache for a configured length of time.

Configuring an OCSP Responder

Configuring OCSP involves adding an OCSP responder, binding the OCSP responder to a certification authority (CA) certificate, and binding the certificate to an SSL virtual server. If you need to bind a different certificate to an OCSP responder that has already been configured, you need to first unbind the responder and then bind the responder to a different certificate.

To add an OCSP responder by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure OCSP and verify the configuration:

- add ssl ocspResponder <name> -**url** <URL> [-**cache** (ENABLED | DISABLED)[-**cacheTimeout** <positive_integer>]] [-**batchingDepth** <positive_integer>][-**batchingDelay** <positive_integer>] [-**resptimeout** <positive_integer>] [-**responderCert** <string> | -**trustResponder**] [-**producedAtTimeSkew** <positive_integer>][-**signingCert** <string>][-**useNonce** (YES | NO)][**-insertClientCert**(YES | NO)]
- bind ssl certKey [<certkeyName>] [-**ocspResponder** <string>] [-**priority** <positive_integer>]
- bind ssl vserver <vServerName>@ (-**certkeyName** <string> (CA [-**ocspCheck** (Mandatory | Optional)]))
- show ssl ocspResponder [<name>]

Example

```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/ocsp/" -cache ENABLED -cacheTime
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory

sh ocspResponder ocsp_responder1
1)Name: ocsp_responder1
URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
Caching: Enabled        Timeout: 30 minutes
Batching: 8 Timeout: 100 mS
HTTP Request Timeout: 100mS
Request Signing Certificate: sign_cert
Response Verification: Full, Certificate: responder_cert
ProducedAt Time Skew: 300 s
```

```
Nonce Extension: Enabled
 Client Cert Insertion: Enabled
Done

show certkey ca_cert
Name: ca_cert     Status: Valid,   Days to expiration:8907
Version: 3
…
1)  VServer name: vs1      CA Certificate
1)  OCSP Responder name: ocsp_responder1     Priority: 1
Done

sh ssl vs vs1
Advanced SSL configuration for VServer vs1:
DH: DISABLED
…
1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
1) Cipher Name: DEFAULT
  Description: Predefined Cipher Alias
Done
```

To modify an OCSP responder by using the NetScaler command line

You cannot modify the responder name. All other parameters can be changed using the set ssl ocspResponder command.

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- set ssl ocspResponder <name> [-**url** <URL>] [-**cache** (ENABLED | DISABLED)] [-**cacheTimeout** <positive_integer>] [-**batchingDepth** <positive_integer>] [-**batchingDelay** <positive_integer>] [-**resptimeout** <positive_integer>] [-**responderCert** <string> | -**trustResponder**][-**producedAtTimeSkew** <positive_integer>][-**signingCert** <string>] [-**useNonce** (YES | NO)]
- unbind ssl certKey [<certkeyName>] [-**ocspResponder** <string>]
- bind ssl certKey [<certkeyName>] [-**ocspResponder** <string>] [-**priority** <positive_integer>]
- show ssl ocspResponder [<name>]

Parameters for configuring an OCSP Responder

name (Name)

The name of the OCSP responder.

url (URL)

The URL of the OCSP responder.

Maximum length: 128.

cache (Cache)

Enable/disable caching of OCSP.

Possible values: ENABLED, DISABLED. Default: DISABLED.

cacheTimeout (Time-out)

OCSP cache timeout, in minutes. If none is specified, the timeout provided in the OCSP response is used.

Range: 1–1440. Default: 60.

batchingDepth (Batching Depth)

Maximum number of client certificates to batch into one OCSP request. Value of 1 signifies that each request is queried independently. Range: 1–8. Default: 1.

batchingDelay (Batching Delay)

Time, in milliseconds, to wait to accumulate OCSP requests.

Range: 0–10000. Default: 1.

resptimeout (Request Time-out)

Time, in milliseconds, to wait for an OCSP response. This is a mandatory parameter. When this time elapses, an error message appears or the transaction is forwarded, depending on the settings on the virtual server. Includes batchingDelay time.

Range: 0–120000. Default: 2000.

responderCert (Certificate)

responderCert specifies the certificate used to validate OCSP responses. trustResponder specifies that responses will not be verified.

producedAtTimeSkew (Produced At Time Skew)

Specifies the time, in seconds, for which the NetScaler waits before considering the response as invalid. The response is considered invalid if the ProducedAt time stamp in the OCSP response exceeds or precedes the current NetScaler clock time by the amount of time specified. Range: 0–86400. Default: 300.

signingCert (Signing Certificate)

Certificate-key pair used to sign OCSP requests. If this parameter is not set, the requests are not signed. Maximum value: 32.

insertClientCert (Client Certificate Insertion)

Include the complete client certificate in the OCSP request. Possible values: YES, NO. Default: NO.

useNonce (Nonce)

Enables the OCSP nonce extension, which is designed to prevent replay attacks.

certKey

The name of the certificate-key pair to bind.

ocspResponder

The name of the OCSP responder to which the certificate-key pair is bound to.

priority

Priority of the OCSP responder.

Range: 0–64000.

sslvserver

The name of the SSL virtual server that the certificate key is bound to.

certkeyname

The name of the certificate.

CA

CA certificate.

ocspCheck

OCSP check is mandatory or optional.

Note: When both OCSP and CRL check are set to optional, OCSP check is used by default. However, if a usable OCSP responder is not available, CRL check is used.

To configure an OCSP responder by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **OCSP Responder**.
2. In the details pane, do one of the following:
   - To create a new responder, click **Add**.
   - To modify an existing responder, select the responder, and then click **Open**.
3. In the **Create OCSP Responder** or **Configure OCSP Responder** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an OCSP Responder" as shown:
   - Name*
   - URL*
   - Cache
   - Time-out
   - Batching—To enable batching of OCSP requests, select this check box.
   - Batching Depth
   - Batching Delay
   - Trust Responses—To disable signature checks by the OCSP responder, select this check box.
   - Certificate
   - Produced At Time Skew
   - Request Time-out
   - Signing Certificate
   - Nonce
   - Client Certificate Insertion

   * A required parameter
4. Click **Create** or **OK**, and then click **Close**.
5. In the **OCSP Responder** pane, click the responder that you just configured and verify that the settings displayed at the bottom of the screen are correct.
6. In the navigation pane, click **Certificates**.
7. In the details pane, select a certificate and click **OCSP Bindings**.

8. In the **OCSP Binding Details for certificate:certkey** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an OCSP Responder" as shown:

    - OCSP Responder Name—ocspResponder. If an OCSP responder is not already bound to the certificate-key pair, click **Insert OCSP Responder** and select a name from the **OCSP Responder Name** drop-down list.

    - Priority—priority
9. To bind a different certificate-key pair, click **Unbind OCSP Responder**, and then click **Insert OCSP Responder** and select a name from the **OCSP Responder Name** drop-down list. Verify that the settings displayed at the bottom of the screen are correct.
10. Click **OK**.
11. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
12. Select the virtual server to bind the certificate key to, and click **Open**.
13. In the **Configure Virtual Server (SSL Offload)** dialog box, click **SSL Settings**.
14. In the **Available** pane, select a certificate.
15. In the **Add** drop-down list select **As CA**.
16. To make OCSP check mandatory, in the **Configured** pane, in the **Check** drop-down list, select **OCSP Mandatory**.
17. Click **OK**.

Configuring Client Authentication

In a typical SSL transaction, the client that is connecting to a server over a secure connection checks the validity of the server by checking the server's certificate before initiating the SSL transaction. In some cases, however, you might want to configure the server to authenticate the client that is connecting to it.

With client authentication enabled on an SSL virtual server, the NetScaler appliance asks for the client certificate during the SSL handshake. The appliance checks the certificate presented by the client for normal constraints, such as the issuer signature and expiration date.

Note: For the NetScaler to verify issuer signatures, the certificate of the CA that issued the client certificate must be installed on the NetScaler and bound to the virtual server that the client is transacting with.

If the certificate is valid, the NetScaler allows the client to access all secure resources. But if the certificate is invalid, the NetScaler drops the client request during the SSL handshake.

The NetScaler verifies the client certificate by first forming a chain of certificates, starting with the client certificate and ending with the root CA certificate for the client (for example, VeriSign). The root CA certificate may contain one or more intermediate CA certificates (if the client certificate is not directly issued by the root CA).

Before you enable client authentication on the NetScaler, make sure that a valid client certificate is installed on the client. Then, enable client authentication for the virtual server that will handle the transactions. Finally, bind the certificate of the CA that issued the client certificate to the virtual server on the NetScaler.

Providing the Client Certificate

Before you configure client authentication, a valid client certificate must installed on the client. A client certificate includes details about the specific client system that will create secure sessions with the NetScaler appliance. Each client certificate is unique and should be used by only one client system.

Whether you obtain the client certificate from a CA, use an existing client certificate, or generate a client certificate on the NetScaler appliance, you must convert the certificate to the correct format. On the NetScaler, certificates are stored in either the PEM or DER format and must be converted to PKCS#12 format before they are installed on the client system. After converting the certificate and transferring it to the client, system, make sure that it is installed on that system and configured for the client application that will be part of the SSL transactions (for example, the web browser).

For instructions on how to convert a certificate from PEM or DER format to PKCS#12 format, see Converting SSL Certificates for Import or Export.

For instructions on how to generate a client certificate, see Generating Self-Signed Certificates.

Enabling Client-Certificate-Based Authentication

By default, client authentication is disabled on the NetScaler appliance, and all SSL transactions proceed without authenticating the client. You can configure client authentication to be either optional or mandatory as part of the SSL handshake.

If client authentication is optional, the NetScaler requests the client certificate but proceeds with the SSL transaction even if the client presents an invalid certificate. If client authentication is mandatory, the NetScaler terminates SSL the handshake if the SSL client does not provide a valid certificate.

Caution: Citrix recommends that you define proper access control policies before changing client-certificate-based authentication check to optional.

Note: Client authentication is configured for individual SSL virtual servers, not globally.

To enable client-certificate-based authentication by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable the client-certificate-based authentication and verify the configuration:

- set ssl vserver <vServerName> [-**clientAuth** (ENABLED | DISABLED)] [-**clientCert** (MANDATORY | OPTIONAL)]
- **show ssl vserver** <vServerName>

Example

```
> set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
 Done
> show ssl vserver vssl

        Advanced SSL configuration for VServer vssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: ENABLED    Client Cert Required: Mandatory
```

```
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      CertKey Name: sslckey   Server Certificate

1)      Policy Name: client_cert_policy  Priority: 0

1)      Cipher Name: DEFAULT
       Description: Predefined Cipher Alias
 Done
```

Parameters for configuring client certificate-based authentication

vServerName

The name of the NetScaler virtual server through which the client will access a physical server.

clientAuth

Enable or disable client authentication. Possible values: ENABLED, DISABLED. Default: DISABLED.

clientCert

Type of client authentication. Possible values: MANDATORY, OPTIONAL.

To enable client-certificate-based authentication by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to configure client certificate-based authentication, and then click **Open**.
3. Click the **SSL Settings** tab, and then click **SSL Parameters**.
4. In the **Others** group, select the **Client Authentication** check box.
5. In **Client Certificate**, select **Mandatory**.

   Note: To configure optional client authentication in **Client Certificate**, click **Optional**.
6. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**. The virtual server is now configured for client authentication.

Binding CA Certificates to the Virtual Server

A CA whose certificate is present on the NetScaler appliance must issue the client certificate used for client authentication. You must bind this certificate to the NetScaler virtual server that will carry out client authentication.

You must bind the CA certificate to the SSL virtual server in such a way that the NetScaler can form a complete certificate chain when it verifies the client certificate. Otherwise, certificate chain formation fails and the client is denied access even if its certificate is valid.

You can bind CA certificates to the SSL virtual server in any order. The NetScaler forms the proper order during client certificate verification.

For example, if the client presents a certificate issued by **CA_A**, where **CA_A** is an intermediate CA whose certificate is issued by **CA_B**, whose certificate is in turn issued by a trusted root CA, **Root_CA**, a chain of certificates that contain all three of these certificates must be bound to the virtual server on the NetScaler.

For instructions on binding one or more certificates to the virtual server, see Binding the Certificate-key Pair to the SSL Based Virtual Server.

For instructions on creating a chain of certificates, see Creating a Chain of Certificates.

Customizing the SSL Configuration

Once your basic SSL configuration is operational, you can customize some of the parameters that are specific to the certificates being used in SSL transactions. You can also enable and disable session reuse and client authentication, and you can configure redirect responses for cipher and SSLv2 protocol mismatches.

You can also customize SSL settings for two NetScaler appliances in a High Availability configuration, and you can synchronize settings, certificates and keys across those appliances.

These settings will depend on your network deployment and the type of clients you expect will connect to your servers.

Configuring Diffie-Hellman (DH) Parameters

If you are using ciphers on the NetScaler that require a DH key exchange to set up the SSL transaction, enable DH key exchange on the NetScaler and configure other settings based on your network.

For details on how to enable DH key exchange, see Generating a Diffie-Hellman (DH) Key.

To configure DH Parameters by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure DH parameters and verify the configuration:

- set ssl vserver <vserverName> -**dh** <Option> -**dhCount** <RefreshCountValue> -**filepath** <string>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.cert -dhCount 1000
  Done
> show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: ENABLED
        Ephemeral RSA: ENABLED         Refresh Count: 1000
        Session Reuse: ENABLED         Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
```

Parameters for configuring DH parameters

dh (Enable DH Param)

Enable or disable DH key exchange. Possible values: ENABLED, DISABLED. Default: DISABLED.

dhCount (Refresh Count)

The number of interactions, between the client and the NetScaler, after which the DH private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). Possible values: 0, or a number greater than 500. Default: 0.

dhFile (File Path)

The absolute path and file name of the DH parameter file to be installed. The default path is /nsconfig/ssl.

To configure DH Parameters by using the NetScaler command line

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize SSL settings, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring DH parameters” as shown:
   - Enable DH Param*
   - Refresh Count
   - File Path*

   * A required parameter
5. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**. The DH parameters are now configured.

Configuring Ephemeral RSA

Ephemeral RSA allows export clients to communicate with the secure server even if the server certificate does not support export clients (1024-bit certificate). If you want to prevent export clients from accessing the secure web object and/or resource, you need to disable ephemeral RSA key exchange.

By default, this feature is enabled on the NetScaler appliance, with the refresh count set to zero (infinite use).

Note:

The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted but reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the system restarts.

To configure Ephemeral RSA by using the NetScaler command

At the NetScaler command prompt, type the following commands to configure ephemeral RSA and verify the configuration:

- set ssl vserver < vServerName > -**eRSA** (enabled | disabled) -**eRSACount** < positive_integer >
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
 Done
> show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
```

```
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
```

Parameters for configuring Ephemeral RSA

eRSA (Enable Ephemeral RSA)

The state of Ephemeral RSA key exchange support for the SSL virtual server.

Possible values: ENABLED, DISABLED

Default value: ENABLED

eRSACount (Refresh Count)

The refresh count for the re-generation of RSA public-key and private-key pair. Zero means infinite usage (no refresh)

Note:

The '-eRSA' argument must be enabled if this argument is specified.

Default value: 0

Minimum value: 0

Maximum value: 65534

To configure Ephemeral RSA by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize SSL settings, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring Ephemera RSA” as shown:
   - Enable Ephemeral RSA*
   - Refresh Count*

   * A required parameter
5. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**. The ephemeral RSA parameters are now configured.

Configuring Session Reuse

For SSL transactions, establishing the initial SSL handshake requires CPU-intensive public key encryption operations. Most handshake operations are associated with the exchange of the SSL session key (client key exchange message). When a client session is idle for some time and is then resumed, the SSL handshake is typically conducted all over again. With session reuse enabled, session key exchange is avoided for session resumption requests received from the client.

Session reuse is enabled on the NetScaler appliance by default. Enabling this feature reduces server load, improves response time, and increases the number of SSL transactions per second (TPS) that can be supported by the server.

To configure session reuse by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure session reuse and verify the configuration:

- set ssl vserver <vServerName> -**sessReuse** < (ENABLED | DISABLED)> -**sessTimeout** < positive_integer>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
 Done

> show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      CertKey Name: Auth-Cert-1       Server Certificate
```

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done

Parameters for configuring Session Reuse

sessReuse (Enable Session Reuse)

Enable or disable the Session Reuse feature on the NetScaler appliance. Possible values: ENABLED, DISABLED. Default: ENABLED.

sessTimeout (Time-out)

Time in seconds up to which the session should be kept active. Any session resumption request received after the time out period will require a fresh SSL handshake and establishment of a new SSL session.

To configure session reuse by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize SSL settings, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring Session Reuse” as shown:
   - Enable Session Reuse*
   - Time-out

   * A required parameter
5. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**.

Configuring Cipher Redirection

During the SSL handshake, the SSL client (usually a web browser) announces the suite of ciphers that it supports, in the configured order of cipher preference. From that list, the SSL server then selects a cipher that matches its own list of configured ciphers.

If the ciphers announced by the client do not match those configured on the SSL server, the SSL handshake fails, and the failure is announced by a cryptic error message displayed in the browser. These messages rarely mention the exact cause of the error.

With cipher redirection, you can configure an SSL virtual server to deliver accurate, meaningful error messages when an SSL handshake fails. When SSL handshake fails, the NetScaler appliance redirects the user to a previously configured URL or, if no URL is configured, displays an internally generated error page.

To configure cipher redirection by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure cipher redirection and verify the configuration:

- set ssl vserver <vServerName> -**cipherRedirect** < ENABLED | DISABLED> -**cipherURL** < URL>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://redirectURl
 Done
> show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: ENABLED        Redirect URL: http://redirectURl
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
```

1) CertKey Name: Auth-Cert-1 Server Certificate

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done

Parameters for configuring Cipher Redirection

vServerName

The name of the SSL based virtual server that you are configuring cipher redirection for.

cipherRedirect (Enable Cipher Redirect)

Enable or disable redirection based on cipher mismatch between the client and the NetScaler. Possible values: ENABLED, DISABLED. Default: DISABLED.

cipherURL (Redirect URL)

The URL of the page to which the client must be redirected in case of a cipher mismatch. This is typically a page that has a clear explanation of the error or an alternate location that the transaction can continue from.

To configure cipher redirection by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize SSL settings, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring cipher redirection” as shown:
   - Enable Cipher Redirect
   - Redirect URL
5. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**. The NetScaler is now configured to redirect clients in case of a cipher suite mismatch.

Configuring SSLv2 Redirection

For an SSL transaction to be initiated, and for successful completion of the SSL handshake, the server and the client should agree on an SSL protocol that both of them support. If the SSL protocol version supported by the client is not acceptable to the server, the server does not go ahead with the transaction, and an error message is displayed.

You can configure the server to display a precise error message (user-configured or internally generated) advising the client on the next action to be taken. Configuring the server to display this message requires that you set up SSLv2 redirection.

To configure SSLv2 redirection by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure SSLv2 redirection and verify the configuration:

- set ssl vserver <vServerName> [-**sslv2Redirect** (ENABLED | DISABLED) [-**sslv2URL** <URL>]]
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -sslv2Redirect ENABLED -sslv2URL http://sslv2URL
 Done
> show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED         Refresh Count: 1000
        Session Reuse: ENABLED         Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: ENABLED Redirect URL: http://sslv2URL
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      CertKey Name: Auth-Cert-1       Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
```

Done

Parameters for configuring SSLv2 redirection

vServerName

The name of the SSL based virtual server that you are configuring SSLv2 redirection for.

sslv2Redirect (Enable SSLv2 Redirect)

Enable or disable redirection based on the SSL protocol mismatch between the client and the NetScaler. Possible values: ENABLED, DISABLED. Default: DISABLED.

sslv2URL (SSLv2 URL)

The URL of the page to which the client must be redirected in case of a protocol mismatch. This is typically a page that has a clear explanation of the error, or an alternative location from which the transaction can continue.

To configure SSLv2 redirection by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize SSL settings, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring SSLv2 redirection” as shown:
   - Enable SSLv2 Redirect
   - SSLv2 URL
5. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**. The NetScaler is now configured to redirect clients that only support SSLv2 protocol.

Configuring SSL Protocol Settings

The NetScaler appliance supports the SSLv2, SSLv3, and TLSv1 protocols. Each of these can be set on the appliance as required by your deployment and the type of clients that will connect to the NetScaler.

To configure SSL protocol support by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure SSL protocol support and verify the configuration:

- set ssl vserver <vServerName> -**tls1** < ENABLED | DISABLED> -**ssl2** < ENABLED | DISABLED> -**ssl3** < ENABLED | DISABLED>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -tls1 ENABLED
 Done
> show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      CertKey Name: Auth-Cert-1       Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
```

Parameters for configuring SSL protocol settings

vServerName

The name of the SSL based virtual server for which you are configuring SSL protocol settings.

tls1 (TLSv1)

Enable or disable support for the TLSv1 protocol on the NetScaler appliance. Possible values: ENABLED, DISABLED. Default: ENABLED.

ssl2 (SSLv2)

Enable or disable support for the SSLv2 protocol on the NetScaler appliance. Possible values: ENABLED, DISABLED. Default: DISABLED.

ssl3 (SSLv3)

Enable or disable support for the SSLv3 protocol on the NetScaler appliance. Possible values: ENABLED, DISABLED. Default: ENABLED.

To configure SSL protocol support by using the configuration Utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize SSL settings, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, in the **SSL Protocol** group, select any of the following protocol options that you want to enable:
   - TLSv1
   - SSLv3
   - SSLv2
5. Click **OK**, and in the **Configure Virtual Server (SSL Offload)** dialog box, click **OK**. The NetScaler now supports the protocol that you enabled.

Configuring Advanced SSL Settings

Advanced customization of your SSL configuration addresses specific issues. You can use the set ssl parameter command or the configuration utility to specify the following:

- Quantum size to be used for SSL transactions.
- CRL memory size.
- OCSP cache size.
- Deny SSL renegotiation.
- Set the PUSH flag for decrypted, encrypted, or all records.
- Drop requests if the client initiates the handshake for one domain and sends an HTTP request for another domain.
- Set the time after which encryption is triggered.

  Note: The time that you specify applies only if you use the set ssl vserver command or the configuration utility to set timer-based encryption.

To configure advanced SSL settings by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure advanced SSL settings and verify the configuration:

- set ssl parameter [-**quantumSize** <quantumSize>] [-**crlMemorySizeMB** <positive_integer>] [-**strictCAChecks** (YES | NO)] [-**sslTriggerTimeout** <positive_integer>] [-**sendCloseNotify** (YES | NO)] [-**encryptTriggerPktCount** <positive_integer>] [-**denySSLReneg** <denySSLReneg>] [-**insertionEncoding** (Unicode|UTF-8)] [-**ocspCacheSize** <positive_integer>][- **pushFlag** <positive_integer>] [- **dropReqWithNoHostHeader** (YES | NO)] [-**pushEncTriggerTimeout** <positive_integer>]
- show ssl parameter

Example

```
> set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks no -sslt
iggerTimeout 100 -sendClosenotify no -encryptTriggerPktCount 45 -denySSLReneg no
-insertionEncoding unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES  -pushEncTriggerT
```

```
 Done

> show ssl parameter
Advanced SSL Parameters
-----------------------
      SSL quantum size:              8 kB
      Max CRL memory size:           256 MB
      Strict CA checks:              NO
      Encryption trigger timeout     100 mS
      Send Close-Notify              NO
      Encryption trigger packet count:       45
      Deny SSL Renegotiation         NO
      Subject/Issuer Name Insertion Format:   Unicode
      OCSP cache size:       10 MB
        Push flag:     0x3 (On every decrypted and encrypted record)
                       Strict Host Header check for SNI enabled SSL sessions:         YES
                        PUSH encryption trigger timeout 100 ms
 Done
```

Parameters for configuring advanced SSL settings

quantumSize (SSL quantum size (Kbytes))

SSL quantum size to be used for SSL transactions on the appliance.

Possible values: 4096, 8192, 16384. Default: 8192.

crlMemorySizeMB (Max CRL memory size (Mbytes))

Maximum memory size to be used for certificate revocation lists.

Minimum value: 10. Maximum value: 1024. Default: 256.

strictCAChecks (Strict CA checks)

Enable strict CA certificate checks on the appliance.

Possible values: YES, NO. Default: NO.

sslTriggerTimeout (Encryption trigger timeout (10 mS ticks))

Encryption trigger timeout value, in milliseconds.

Note: There may be a delay of up to 10 ms from the specified timeout value before the packet is pushed into the queue.

Minimum value: 1. Maximum value: 200. Default: 100.

sendCloseNotify (Send Close-Notify)

Enable sending an SSL Close-Notify message to the client at the end of a transaction.

Possible values: YES, NO. Default: YES.

encryptTriggerPktCount (Encryption trigger packet count)

Number of queued packets that force encryption to occur.

Minimum value: 10. Maximum value: 50. Default: 45.

denySSLReneg (Deny SSL Renegotiation)

Deny renegotiation in specified circumstances. Possible values:

NO—Allow SSL renegotiation.

FRONTEND_CLIENT—Deny secure and nonsecure SSL renegotiation initiated by the client.

FRONTEND_CLIENTSERVER—Deny secure and nonsecure SSL renegotiation initiated by the client and by the NetScaler (during policy-based clientAuth).

ALL—Deny secure and nonsecure SSL renegotiation for the above two cases and for server initiated renegotiation.

NONSECURE—Deny nonsecure SSL renegotiation, to address the vulnerability described in RFC 5746.

Note: The option is supported only on NetScaler 9.3.e.

Default: NO.

insertionEncoding (Encoding type)

Encoding method used to insert the subject or issuer's name in HTTP requests to backend servers.

Possible values: Unicode, UTF-8. Default: Unicode.

ocspCacheSize (OCSP cache size(Mbytes))

Size, per packet engine, in megabytes, of the OCSP cache. The actual maximum value for this value is clamped at 10% of packet engine memory. Maximum packet engine memory is 4GB. Therefore, if you have enough memory to give all packet engines 4GB of memory, the maximum value here would be approximately 410 MB.

Minimum value: 0. Maximum value: 512. Default: 10.

pushFlag (PUSH Flag Insertion)

Insert PUSH flag into decrypted, encrypted, or all records. If the PUSH flag is set to a value other than 0, the buffered records are forwarded on the basis of the value of the PUSH flag. Possible values:

0—Auto (PUSH flag is not set).

1—Insert PUSH flag into every decrypted record.

2—Insert PUSH flag into every encrypted record.

3—Insert PUSH flag into every decrypted and encrypted record.

Possible values: 0, 1, 2, 3. Default: 0.

dropReqWithNoHostHeader (Drop requests for SNI enabled SSL sessions if Host header is absent)

Host header check for SNI enabled sessions. If this check is enabled and the HTTP request does not contain the Host header for SNI enabled session, the request is dropped.

Possible values: YES, NO. Default: NO.

pushEncTriggerTimeout (PUSH encryption trigger timeout (msec))

Encryption trigger timeout value. The timeout value that is applied when timer option is specified in the set ssl vserver **-pushEncTrigger** command.

Minimum value: 1. Maximum value: 200. Default: 1.

To configure advanced SSL settings by using the configuration utility

1. In the navigation pane, click **SSL**.
2. In the details pane, under **Settings**, click **Change advanced SSL settings**.
3. In the **Change advanced SSL settings** dialog box, set the following parameters:
   - SSL quantum size (Kbytes)
   - Max CRL memory size (Mbytes)
   - Encryption trigger timeout (10 mS ticks)
   - Encryption trigger packet count
   - Deny SSL Renegotiation
   - OCSP cache size(Mbytes)
   - Encoding type
   - PUSH encryption trigger timeout (msec)
   - Strict CA checks
   - Send Close-Notify
   - PUSH Flag Insertion
   - Drop requests for SNI enabled SSL sessions if Host header is absent
4. Click **OK**. The parameters you selected are now enabled on the NetScaler.

PUSH Flag-Based Encryption Trigger Mechanism

The encryption trigger mechanism that is based on the PSH TCP flag now enables you to do the following:

- Merge consecutive packets in which the PSH flag is set into a single SSL record, or ignore the PSH flag.
- Perform timer-based encryption, in which the time-out value is set globally by using the set ssl parameter **-pushEncTriggerTimeout** <positive_integer> command.

To configure PUSH flag-based encryption by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure PUSH flag-based encryption and verify the configuration:

- set ssl vserver <vServerName> [**-pushEncTrigger** <pushEncTrigger>]
- show ssl vserver

Example

```
Advanced SSL configuration for VServer v1:
        DH: DISABLED
        Ephemeral RSA: ENABLED                  Refresh Count: 0
        Session Reuse: ENABLED                  Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        SSLv2: DISABLED         SSLv3: ENABLED          TLSv1: ENABLED
        Push Encryption Trigger: Always
```

Parameters for configuring PUSH flag-based encryption

pushEncTrigger

Trigger encryption based on the value of the PUSH flag.

Possible values:

- **Always.** Any PUSH packet triggers encryption.
- **Ignore.** Ignore PUSH packet for triggering encryption.
- **Merge.** For a consecutive sequence of PUSH packets, the last PUSH packet triggers encryption.
- **Timer.** PUSH packet triggering encryption is delayed by the time defined in the set ssl parameter command. Possible values: Always, Ignore, Merge, Timer. Default: Always.

To configure PUSH flag-based encryption by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. Select the virtual server for which you want to customize PUSH-flag based encryption, and then click **Open**.
3. On the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Configure SSL Params** dialog box, select a value for the **PUSH Encryption Trigger** parameter. For descriptions of the values, see “Parameters for configuring PSH flag-based encryption.”
5. Click **OK** and, in the **Configure Virtual Server (SSL Offload)** dialog box, again click **OK**. The PSH flag-based encryption trigger is now configured.

Synchronizing Configuration Files in a High Availability Setup

In a high availability (HA) set up, the primary NetScaler appliance in the HA pair automatically synchronizes with the secondary appliance in the pair. In the synchronization process, the secondary copies the primary's /nsconfig/ssl/ directory, which is the default location for storing the certificates and keys for SSL transactions. Synchronization occurs at one-minute intervals and every time a new file is added to the directory.

To synchronize files in a high availability setup by using the NetScaler command line

At the NetScaler command prompt, type the following command:

sync HA files [<Mode> ...]

Example

sync HA files SSL

Parameters for synchronizing files in a high availability set up

mode

The type of synchronization to be performed. The following options are available:

- **All.** Synchronizes all data.
- **bookmarks.** Synchronizes all Access Gateway bookmarks.
- **SSL.** Synchronizes all the SSL certificates and keys that are defined on the NetScaler appliance.

To synchronize files in a high availability setup by using the configuration utility

1. In the navigation pane, click **SSL**.
2. In the details pane, under **Tools**, click **Start file synchronization**.
3. In the **Start file synchronization** dialog box, in the **Mode** drop-down list, select the appropriate type of synchronization (for example, **SSL certificates and Keys**), and then click **OK**.

Managing Server Authentication

Since the NetScaler appliance performs SSL offload and acceleration on behalf of a web server, the appliance does not usually authenticate the Web server's certificate. However, you can authenticate the server in deployments that require end-to-end SSL encryption.

In such a situation, the NetScaler becomes the SSL client, carries out a secure transaction with the SSL server, verifies that a CA whose certificate is bound to the SSL service has signed the server certificate, and checks the validity of the server certificate.

To authenticate the server, you must first enable server authentication and then bind the certificate of the CA that signed the server's certificate to the SSL service on the NetScaler. When binding the certificate, you must specify the bind as CA option.

To enable (or disable) server certificate authentication by using the NetScaler command line

At the NetScaler command prompt, type the following commands to enable server certificate authentication and verify the configuration:

- set ssl service <serviceName> -**serverAuth** < ENABLED | DISABLED
- show ssl service <serviceName>

Example

```
> set ssl service ssl-service-1 -serverAuth ENABLED
 Done
> show ssl service ssl-service-1

      Advanced SSL configuration for Back-end SSL Service ssl-service-1:
      DH: DISABLED
      Ephemeral RSA: DISABLED
      Session Reuse: ENABLED        Timeout: 300 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      Server Auth: ENABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)     Cipher Name: ALL
      Description: Predefined Cipher Alias
 Done
```

Parameters for enabling or disabling server certificate authentication

serviceName

The name of the service for which you are configuring server certificate authentication.

serverAuth

Enable or disable server authentication. This is used when you configure end-to-end SSL encryption to verify the authenticity of the server.

To enable (or disable) server certificate authentication by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Services**.
2. Select the service for which you want to enable server authentication, and then click **Open**.
3. In **Configure Service** dialog box, on the **SSL Settings** tab, click **SSL Parameters**.
4. In the **Others** group, select **Server Authentication**.
5. Click **OK**. Server authentication is now enabled for the service.

To bind the CA certificate to the service by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind the CA certificate to the service and verify the configuration:

- bind ssl service <serviceName> -**certkeyName** < string> -**CA**
- show ssl service <serviceName>

Example

```
> bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
 Done
> show ssl service ssl-service-1

        Advanced SSL configuration for Back-end SSL Service ssl-service-1:
        DH: DISABLED
        Ephemeral RSA: DISABLED
```

```
        Session Reuse: ENABLED          Timeout: 300 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        Server Auth: ENABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      CertKey Name: samplecertkey     CA Certificate          CRLCheck: Optional

1)      Cipher Name: ALL
        Description: Predefined Cipher Alias
 Done
```

Parameters for managing server authentication

serviceName

The name of the service for which server authentication is configured.

certkeyName

The name of the certificate key pair that is bound to the SSL service.

CA

Specifies that the certificate-key pair being bound belongs to a Certificate Authority that has signed the server certificate.

To bind the CA certificate to the service by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Services**.
2. Select the service for which you want to enable server authentication, and then click **Open**.
3. In **Configure Service** dialog box, under **Available Certificates**, select the CA certificate you want to bind, and then click **Add as CA**.
4. Click **OK**. The CA certificate is now bound to the SSL service.

Configuring User-Defined Cipher Groups on the NetScaler Appliance

A cipher group is a set of cipher suites that you bind to an SSL virtual server or service on the NetScaler appliance. A cipher suite comprises a protocol, a key exchange (Kx) algorithm, an authentication (Au) algorithm, an encryption (Enc) algorithm, and a message authentication code (Mac) algorithm. Your appliance ships with a predefined set of cipher groups. When you create an SSL virtual server or service, the Default cipher group is automatically bound to it. In addition, you can create a user-defined cipher group and bind it to an SSL virtual server. To create a user-defined cipher group, you select one or more groups from the available cipher groups. If you specify a cipher alias or a cipher group, all the ciphers in the cipher alias or group are added to the user-defined cipher group. You can also add individual ciphers (cipher suites) to a user-defined group. However, you cannot modify a predefined cipher group.

Note: The free NetScaler VPX virtual appliance supports only the DH cipher group.

To add a cipher group or add ciphers to an existing group by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add a cipher group, or to add ciphers to a previously created group, and verify the settings:

- add ssl cipher <userDefCipherGroupName> (<cipherAliasName> | <cipherName> | <cipherGroupName>)
- show ssl cipher (<cipherAliasName> | <cipherName> | <cipherGroupName>)]

Example

```
> add ssl cipher test SSLv2
Done
> show ssl cipher test
1) Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
2) Cipher Name: SSL2-DES-CBC3-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
3) Cipher Name: SSL2-RC2-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC2(128) Mac=MD5
4) Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5
5) Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(64) Mac=MD5
```

```
6) Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 Export
7) Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 Export
Done
```

To remove a cipher group by using the NetScaler command line

Note: You cannot remove a built-in cipher group.

At the NetScaler command prompt, type the following command to remove a user-defined cipher group:

rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]

Example

```
> rm ssl cipher test
Done
```

Parameters for configuring a user-defined cipher group

userDefCipherGroupName (Cipher Group Name)

The name of the user-defined cipher group. If the cipher group does not exist on the appliance, a new group with the specified name is created, and the ciphers are added to this group. If a group identified by userDefCipherGroupName already exists, the ciphers are added to it. This is a required parameter. Maximum Length: 39 characters.

cipherAliasName/cipherName/cipherGroupName (Available/Configured Cipher Groups/Ciphers)

The individual cipher name(s), a user-defined cipher group, or a predefined (built-in) cipher alias to be added to or removed from the user-defined cipher group. This is a required parameter. Maximum Length: 39 characters.

To configure a user-defined cipher group by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Cipher Groups**.
2. In the details pane, do one of the following:
   - To create a new cipher group, click **Add.**
   - To modify an existing cipher group, select the cipher group, and then click **Open.**
3. If creating a new cipher group, in the **Create Cipher Group** dialog box, in the **Cipher Group Name** box, type a name for the new cipher group.
4. In the **Create Cipher Group** or **View Cipher Group** dialog box, do any or all of the following:
   - Select a cipher group or alias in the **Available Cipher Groups** list and click **Add** to move the group to the **Configured Cipher Groups** list.
   - Select a cipher group or alias in the **Available Cipher Groups** list, then select ciphers from the **Available Ciphers** list, and then click **Add** to move the selected ciphers to the **Configured Ciphers** list.
   - To move a cipher group or cipher from the Configured list to the Available list, select the group or cipher and click **Remove**.
5. Click **Create**, and then click **Close**. If you created a new cipher group, it appears in the **Cipher Groups** pane.

To bind a cipher group to an SSL virtual server, service, or service group by using the NetScaler command line

At the NetScaler command prompt, type:

bind ssl cipher (<vServerName> | <serviceName> | <serviceGroupName>) [- **vServer** | **-service]** <cipherOperation> <cipherAliasName/cipherName/cipherGroupName>

Example

```
bind ssl cipher testv1 ADD SSLv3
Done
bind ssl cipher testv1 REM DH
Done
bind ssl vserver testv1 ORD HIGH
```

Parameters for Binding a Cipher Group to an SSL Virtual Server, Service, or Service Group

vServerName (Name)

The name of the SSL virtual server to which the cipher suite is to be bound. Maximum Length: 127 characters.

vServer

Set the -vServer flag, which specifies that the cipher operation is performed on an SSL virtual server. (The configuration utility sets this parameter transparently.)

serviceName (Service Name)

The name of the SSL service to which the cipher suite is to be bound. Maximum Length: 127 characters.

service

Set the -service flag, which specifies that the cipher operation is performed on an SSL service or service group. (The configuration utility sets this parameter transparently.)

serviceGroupName (Service Group Name)

The name of the SSL service group to which the cipher suite is to be bound. Maximum Length: 127 characters.

cipherOperation (Add/Remove)

The operation that is performed when adding the cipher suite. Possible cipher operations are:

- ADD - Appends the given cipher suite to the existing one configured for the virtual server, service, or service group.
- REM - Removes the given cipher suite from the existing one configured for the virtual server, service, or service group.
- ORD - Overrides the current configured cipher suite for the virtual server, service, or service group with the specified cipher suite.

Possible values: ADD, REM, ORD.

cipherAliasName/cipherName/cipherGroupName

A cipher suite can consist of an individual cipher name, the predefined cipher-alias name, or user-defined cipher group name. This is a required parameter. Maximum Length: 39 characters.

To bind a cipher group to an SSL virtual server, service, or service group by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**, **Services**, or **Service Groups**.
2. In the details pane, select the virtual server, service, or service group to bind the cipher to, and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)**, **Configure Service**, or **Configure Service Group** dialog box, on the **SSL Settings** tab, click **Ciphers**.
4. In the **SSL-Offload - Configure Ciphers**, **Service - Configure Ciphers**, or **Service Group - Configure Ciphers** dialog box, do one or both of the following:
   - To bind a cipher group, select a cipher group or alias from the **Available Cipher Groups** list, and then click **Add**. To unbind a group, select the cipher group or alias from the **Configured Cipher Groups** list, and then click **Remove**.
   - To bind a cipher, select a cipher group or alias from the **Available Cipher Groups** list, then select ciphers from the **Available Ciphers** list, and then click **Add**. To unbind a cipher. select the cipher from the **Configured Ciphers** list, and then click **Remove**.

   Note: To override an existing cipher or cipher group, drag and drop the configured cipher or cipher group to a new location in the **Configured Ciphers** list or the **Configured Cipher Groups** list so that it precedes the cipher or cipher group to be overridden.
5. Click **OK** to close the dialog box, and then click **OK** again.

Configuring SSL Actions and Policies

SSL actions define SSL settings that you can apply to selected connections. You associate an action with one or more polices. Data in client connection requests or responses is compared to a rule specified in the policy, and the action is applied to connections that match the rule (expression). SSL policies use the simpler of two NetScaler expression languages, called *classic expressions*. For a complete description of classic expressions, how they work, and how to configure them manually, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: Users who are not experienced in configuring policies at the NetScaler command line usually find using the configuration utility to be considerably easier.

You bind each policy either globally or to one or more virtual servers, to specify that the policy is used to process all traffic or just the traffic that flows through specific virtual servers. For more information about binding policies, including custom bind points (specific points in the processing of requests and responses) for globally bound policies, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Common uses of SSL actions and policies include per-directory client authentication, support for Outlook web access, and SSL based header insertions. SSL based header insertions contain SSL settings required by a server whose SSL processing has been offloaded to the NetScaler appliance.

Configuring Per-Directory Client Authentication

If you create an action specifying client-side authentication on a per-directory basis, a client identified by a policy associated with the action is not authenticated as part of the initial SSL handshake. Instead, authentication is carried out every time the client wants to access a specific directory on the web server.

For example, if you have multiple divisions in the company, where each division has a folder in which all its files are stored, and you want to know the identity of each client that tries to access files from a particular directory, such as the finance directory, you can enable per-directory client authentication for that directory.

To enable per-directory client authentication, first configure client authentication as an SSL action, and then create a policy that identifies the directory that you want to monitor. When you create the policy, specify your client-authentication action as the action associated with the policy. Then, bind the policy to the SSL virtual server that will receive the SSL traffic.

To create an SSL action to enable client authentication by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL action to enable to client authentication and verify the configuration:

- add ssl action <name> [-**clientAuth** (DOCLIENTAUTH | NOCLIENTAUTH)]
- show ssl action [<name>]

Example

```
> add ssl action ssl-action-1 -clientAuth DOCLIENTAUTH
 Done
> show ssl action ssl-action-1
1)      Name: ssl-action-1
       Client Authentication Action: DOCLIENTAUTH
 Done
```

Parameters for enabling client authentication

name (Name)

The name for the new SSL action. Maximum Length: 127

This is a mandatory argument.

clientAuth (Client Authentication)

Set action to do client certificate authentication or no authentication.

DOCLIENTAUTH: Perform client certificate authentication.

NOCLIENTAUTH: No client certificate authentication.

Possible values: DOCLIENTAUTH, NOCLIENTAUTH

To create an SSL action to enable client authentication by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. On the **Actions** tab, in the details pane, click **Add**.
3. In the **Create SSL Action** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for enabling client authentication” as shown:
   - Name*
   - Client Authentication

   * A required parameter
4. Click **Create**, and then click **Close**.

To create and bind an SSL policy to enable client authentication

SSL policies use the simpler of two NetScaler expressions languages, called *classic expressions*. For a complete description of classic expressions and how to configure them, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To bind the policy, see Binding SSL Policies to a Virtual Server.

Configuring Support for Outlook Web Access

If your SSL configuration is offloading SSL transactions from an Outlook Web Access (OWA) server, you must insert a special header field, FRONT-END-HTTPS: ON, in all HTTP requests directed to the OWA servers. This is required for the OWA servers to generate URL links as https:// instead of http://.

When you enable support for OWA on the NetScaler, the header is automatically inserted into the specified HTTP traffic, and you do not need to configure a specific header insertion. Use SSL policies to identify all traffic directed to the OWA server.

Note: You can enable Outlook Web Access support for HTTP-based SSL virtual servers and services only. You cannot apply it to TCP-based SSL virtual servers and services.

To enable OWA support, first configure OWA support as an SSL action, and then create a policy that identifies the virtual servers or services for which you want to enable OWA support. When you create the policy, specify your OWA support action as the action associated with the policy. Then, bind the policy to the SSL virtual server that will receive the SSL traffic.

To create an SSL action to enable OWA support by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

- add ssl action <name> -**OWASupport** (ENABLED | DISABLED)
- show ssl action [<name>]

Example

```
> add ssl action ssl-action-2 -OWASupport ENABLED
 Done
> show ssl action ssl-action-2
1)      Name: ssl-action-2
        Data Insertion Action:
       OWA Support: ENABLED
 Done
```

Parameters for enabling support for Outlook Web Access

name (Name)

The name for the new SSL action. Maximum Length: 127

This is a mandatory argument.

OWASupport (Outlook Web Access)

The state of Outlook Web-Access support. If the system is in front of an Outlook Web Access (OWA) server, a special header field, 'FRONT-END-HTTPS: ON', needs to be inserted in the HTTP requests going to the OWA server.

To create an SSL action to enable OWA support by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. On the **Actions** tab, in the details pane, click **Add**.
3. In the **Create SSL Action** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for enabling support for Outlook Web Access” as shown:
   - Name*
   - Outlook Web Access

   * A required parameter
4. Click **Create**, and then click **Close**.

   Note: Outlook Web Access support is applicable only for SSL virtual server based configurations and transparent SSL service based configurations and not for SSL configurations with back-end encryption.

To create and bind an SSL policy to enable OWA support

SSL policies use the simpler of two NetScaler expressions languages, called *classic expressions*. For a complete description of classic expressions and how to configure them, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To bind the policy, see Binding SSL Policies to a Virtual Server.

Configuring SSL-based Header Insertion

Because the NetScaler appliance offloads all SSL-related processing from the servers, the servers receive only HTTP traffic. In some circumstances, the server needs certain SSL information. For example, security audits of recent SSL transactions require the client subject name (contained in an X509 certificate) to be logged on the server.

Such data can be sent to the server by inserting it into the HTTP header as a name-value pair. You can insert the entire client certificate, if required, or only the specific fields from the certificate, such as the subject, serial number, issuer, certificate hash, SSL session ID, cipher suite, or the not-before or not-after date used to determine certificate validity.

You can enable SSL-based insertion for HTTP-based SSL virtual servers and services only. You cannot apply it to TCP-based SSL virtual servers and services. Also, client authentication must be enabled on the SSL virtual server, because the inserted values are taken from the client certificate that is presented to the virtual server for authentication.

To configure SSL-based header insertion, first create an SSL action for each specific set of information to be inserted, and then create policies that identify the connections for which you want to insert the information. As you create each policy, specify the action that you want associated with the policy. Then, bind the policies to the SSL virtual servers that will receive the SSL traffic.

To configure an action for SSL-based header insertion by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure an action for SSL-based header insertion and verify the configuration:

- add SSL action <name> -**clientCert** (ENABLED | DISABLED) -**clientHeader** <string> -**clientSerialNumber** (ENABLED | DISABLED) -**clientSerialHeader** <string> **-clientCertSubject** (ENABLED | DISABLED) -certSubjectHeader <string> -**clientCertHash** (ENABLED | DISABLED) -**certHashHeader** <string> -**clientCertIssuer** (ENABLED | DISABLED) -**certIssuerHeader** <string> -**sessionID** (ENABLED | DISABLED) -**sessionIDheader** <string> -**cipher** (ENABLED | DISABLED) -**cipherHeader** <string> -**clientCertNotBefore** (ENABLED | DISABLED) **-certNotBeforeHeader** <string> -**clientCertNotAfter** (ENABLED | DISABLED) -**certNotAfterHeader** <string>

- show ssl action [<name>]

Example

```
> add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X-Client-Cert"
 Done
```

```
> show ssl action Action-SSL-ClientCert
1)      Name: Action-SSL-ClientCert
        Data Insertion Action:
       Cert Header: ENABLED          Cert Tag: X-Client-Cert
 Done
```

Parameters for configuring an SSL-based header insertion action

name (Name)

The name of the SSL-based action for which you are configuring header insertion. Maximum Length: 127

clientCert (Client Certificate)

Insert the entire client certificate into the HTTP header of the request being sent to the Web server. The certificate is inserted in the ASCII (PEM) format. Possible values: ENABLED, DISABLED.

certHeader (Certificate Tag)

The name of the header in which the client certificate is inserted.

clientCertSerialNumber (Client Certificate Serial Number)

Insert the entire client serial number into the HTTP header of the request being sent to the Web server. Possible values: ENABLED, DISABLED

certSerialHeader (Serial Number Tag)

The name of the header in which the client serial number is inserted.

clientCertSubject (Client Certificate Subject (DN))

Insert the client certificate subject also known as the distinguished name (DN) into the HTTP header of the request being sent to the Web server. Possible values: ENABLED, DISABLED

clientSubjectHeader (Subject Tag)

The name of the header in which the client certificate subject is inserted.

clientCertHash (Client Certificate Hash)

Insert the certificate signature (hash) into the HTTP header of the request being sent to the Web server. Possible values: ENABLED, DISABLED

certHashHeader (Hash Tag)

The name of the header in which the client certificate signature (hash) is inserted.

clientCertIssuer (Client Certificate Issuer)

Insert the certificate certificate issuer into the HTTP header of the request being sent to the Web server. Possible values: ENABLED, DISABLED

certIssuerHeader (Issuer Tag)

The name of the header in which the client certificate issuer details are inserted.

sessionID (Session ID)

Insert the SSL session ID into the HTTP header of the request being sent to the Web server. Every SSL connection that the client and the NetScaler share has a unique ID that identifies the specific connection. Possible values: ENABLED, DISABLED.

sessionIDHeader (Session ID Tag)

The tag name to be used while inserting the Session-ID in the HTTP header. Maximum length: 31

cipher (Cipher Suite)

Insert the cipher suite negotiated by the client and the NetScaler for the particular SSL session into the HTTP header of the request being sent to the Web server. The NetScaler will insert the cipher-suite name, SSL protocol, export or non-export string, and cipher strength bit, depending on the type of browser connecting to the SSL virtual server/service (for example, Cipher-Suite: RC4- MD5 SSLv3 Non-Export 128-bit).

cipherHeader (Cipher Tag)

The name of the header in which the name of the cipher suite is inserted.

clientCertNotBefore (Client Certificate Not Before Date)

Insert the date from which the certificate is valid into the HTTP header of the request being sent to the Web server. Every certificate is configured with the date and time from which it is valid, which is contained in this header.

certNotBeforeHeader (Not Before Tag)

The name of the header into which to insert the date and time from which the certificate is valid.

clientCertNotAfter (Client Certificate Not After Date)

Insert the date of expiry of the certificate into the HTTP header of the request being sent to the Web server. Every certificate is configured with the date and time at which the certificate expires, which is contained in this header.

certNotAfterHeader (Not After Tag)

The name of the header in which the certificate's expiry date is inserted.

OWASupport (Outlook Web Access)

If the system is in front of an Outlook Web Access (OWA) server, a special header field, 'FRONT-END-HTTPS: ON', needs to be inserted in the HTTP requests going to the OWA server.

To configure an action for SSL based header insertion by using the NetScaler command line

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. On the **Actions** tab, in the details pane, click **Add**.
3. In the **Create SSL Action** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring an SSL-based header insertion action" as shown:
   - Name*
   - Client Certificate
   - Certificate Tag
   - Client Certificate Serial Number
   - Serial Number Tag
   - Client Certificate Subject (DN)
   - Subject Tag
   - Client Certificate Hash
   - Hash Tag
   - Client Certificate Issuer
   - Issuer Tag
   - Session ID
   - Session ID Tag
   - Cipher Suite
   - Cipher Tag
   - Client Certificate Not Before Date
   - Not Before Tag
   - Client Certificate Not After Date
   - Not After Tag
   - Outlook Web Access

   * A required parameter
4. Click **Create**, and then click **Close**.

To create and bind an SSL policy for SSL based header insertion

SSL policies use the simpler of two NetScaler expressions languages, called *classic expressions*. For a complete description of classic expressions and how to configure them, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To bind the policy, see Binding SSL Policies to a Virtual Server.

Configuring SSL Policies

Policies on the NetScaler help identify specific connections that you want to process. The processing is based on the actions that are configured for that particular policy. Once you create the policy and configure an action for it, you must either bind it to a virtual server on the NetScaler, so that it applies only to traffic flowing through that virtual server, or bind it globally, so that it applies to all traffic flowing through any virtual server configured on the NetScaler.

SSL policies use the simpler of two NetScaler expressions languages, called *classic expressions*. For a complete description of classic expressions, how they work, and how to configure them manually, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

Note: Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

Binding SSL Policies to a Virtual Server

The SSL policies that are configured on the NetScaler appliance need to be bound to a virtual server that intercepts traffic directed to the virtual server. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

You can also bind SSL policies globally or to custom bind points on the NetScaler. For more information about binding policies on the NetScaler, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX128673.

To bind an SSL policy to a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following command to bind an SSL policy to a virtual server and verify the configuration:

- bind ssl vserver <vServerName> -**policyName** <string> [-**priority** <positive_integer>]
- show ssl vserver <vServerName>

Example

```
> bind ssl vserver vs-server -policyName ssl-policy-1 -priority 10
 Done
> show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 80
        Client Auth: DISABLED
        SSL Redirect: ENABLED
        SSL-REDIRECT Port Rewrite: ENABLED
        Non FIPS Ciphers: DISABLED
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Policy Name: ssl-policy-1       Priority: 10

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
```

Done

Parameters for binding SSL policies to a virtual server

vServerName

The name of the SSL virtual server to which the SSL policy needs to be bound.

This is a mandatory argument. Maximum Length: 127

policyName

The name of the SSL policy. Maximum Length: 127

priority

Priority. Minimum value: 0

Maximum value: 64000

To bind an SSL policy to a virtual server by using the configuration utility

1. In the navigation pane, expand **SSL Offload**, and then click **Virtual Servers**.
2. In the details pane, from the list of virtual servers, select the virtual server to which you want to bind the SSL policy, and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box**,** on the **Policies** tab**,** in the details pane, click **Insert Policy**.
4. Under **Policy Name**, select the policy that you want to bind to the virtual server.
5. Click **OK**, and then click **Close**. A message appears in the status bar, stating that the policy has been bound successfully.

Binding SSL Policies Globally

Globally bound policies are evaluated after all policies bound to services, virtual servers, or other NetScaler bind points are evaluated.

To globally bind an SSL policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to bind a global SSL policy and verify the configuration:

- bind ssl global **- policyName** <string> [- **priority** <positive_integer>]
- show ssl global

Example

```
> bind ssl global -policyName Policy-SSL-2 -priority 90
Done
> sh ssl global
1) Name: Policy-SSL-2 Priority: 90
2) Name: Policy-SSL-1 Priority: 100
Done
```

Parameters for globally binding an SSL policy

policyName (Policy Name)

The name of the SSL policy. Maximum Length: 127.

priority (Priority)

A numeric value that indicates when this policy is evaluated relative to others. A lower priority is evaluated before a higher one.

To bind a global SSL policy by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **Bind/Unbind SSL Policies to Global** dialog box, click **Insert Policy**.
4. In the **Policy Name** drop-down list, select a policy.
5. Optionally, drag the entry to a new position in the policy bank to automatically update the priority level.
6. Click **OK**. A message appears in the status bar, stating that the policy has been bound successfully.

Commonly Used SSL Configurations

SSL deployments typically use some version of one or more of the following configurations:

- SSL Offloading with End-to-End Encryption
- Transparent SSL Acceleration
  - Service-based Transparent SSL Acceleration
  - Virtual Server-based Acceleration with a Wildcard IP Address (*:443)
  - SSL VIP-based Transparent Access with End-To-End Encryption
- SSL Acceleration with HTTP on the Front-End and SSL on the Back-End
- SSL Offloading with Other-TCP Protocols
  - SSL_TCP Based Offloading with End-to-End Encryption
  - Backend Encryption for TCP Based Data
- SSL Bridging

Configuring SSL Offloading with End-to-End Encryption

A simple SSL offloading setup terminates SSL traffic (HTTPS), decrypts the SSL records, and forwards the clear text (HTTP) traffic to the back-end web servers. However, the clear text traffic is vulnerable to being spoofed, read, stolen, or compromised by individuals who succeed in gaining access to the back-end network devices or web servers.

You can, therefore, configure SSL offloading with end-to-end security by re-encrypting the clear text data and using secure SSL sessions to communicate with the back-end Web servers.

Additionally, you can configure the back-end SSL transactions so that the NetScaler appliance uses SSL session multiplexing to reuse existing SSL sessions with the back-end web servers, thus avoiding CPU-intensive key exchange (full handshake) operations. This reduces the overall number of SSL sessions on the server, and therefore accelerates the SSL transaction while maintaining end-to-end security.

To configure SSL Offloading with end-to-end encryption, add SSL based services that represent secure servers with which the NetScaler appliance will carry out end-to-end encryption. Then create an SSL based virtual server, and create and bind a valid certificate-key pair to the virtual server. Bind the SSL services to the virtual server to complete the configuration.

For details on adding SSL based services, see Configuring Services.

For details on adding an SSL virtual server, see Configuring an SSL Based Virtual Server.

For details on creating a certificate-key pair, see Adding a Certificate/Key Pair.

For details on binding a certificate-key pair to a virtual server, see Binding the Certificate Key Pair to the SSL Based Virtual Server.

For details on binding services to a virtual server, see Binding Services to the SSL Based Virtual Server.

Example

Create two SSL based services, Service-SSL-1 and Service-SSL-2, with IP addresses 10.102.20.30 and 10.102.20.31 and both using port 443.

Then create an SSL based virtual server, Vserver-SSL-2 with an IP address of 10.102.10.20.

Next, create a certificate-key pair, CertKey-1 and bind it to the virtual server.

Bind the SSL services to the virtual server to complete the configuration.

Table 1. Entities in the SSL Offloading with End-to-End Encryption Example

Entity	Name	Value
SSL Service	Service-SSL-1	10.102.20.30
	Service-SSL-2	10.102.20.31
SSL Based Virtual Server	Vserver-SSL-2	10.102.10.20
Certificate - Key Pair	Certkey-1	

Configuring Transparent SSL Acceleration

Note: You need to enable L2 mode on the NetScaler appliance for transparent SSL acceleration to work.

Transparent SSL acceleration is useful for running multiple applications on a secure server with the same public IP, and also for SSL acceleration without using an additional public IP.

In a transparent SSL acceleration setup, the NetScaler appliance is transparent to the client, because the IP address at which the appliance receives requests is the same as the Web server's IP address.

The NetScaler offloads SSL traffic processing from the Web server and sends either clear text or encrypted traffic (depending on the configuration) to the web server. All other traffic is transparent to the NetScaler and is bridged to the Web server. Therefore, other applications running on the server are unaffected.

There are three modes of transparent SSL acceleration available on the NetScaler:

- Service-based transparent access, where the service type can be SSL or SSL_TCP.
- Virtual server-based transparent access with a wildcard IP address (*:443).
- SSL VIP-based transparent access with end-to-end encryption.

Note: An SSL_TCP service is used for non-HTTPS services (for example SMTPS, and IMAPS).

Service-based Transparent SSL Acceleration

To enable transparent SSL acceleration using the SSL service mode, configure an SSL or an SSL_TCP service with the IP address of the actual back-end Web server. Instead of a virtual server intercepting SSL traffic and passing it on to the service, the traffic is now directly passed on to the service, which decrypts the SSL traffic and sends clear text data to the back-end server.

The service-based mode allows you to configure individual services with a different certificate, or with a different clear text port. Also, you can also select individual services for SSL acceleration.

You can apply service-based transparent SSL acceleration to data that uses different protocols, by setting the clear text port of the SSL service to the port on which the data transfer between the SSL service and the back-end server occurs.

To configure service-based transparent SSL acceleration, first enable both the SSL and the load balancing features. Then create an SSL based service and configure its clear text port. After the service is created, create and bind a certificate-key pair to this service.

For details about SSL, see Enabling SSL Processing. For details about load balancing, see Load Balancing.

For details on adding an SSL based services, see Configuring Services.

For details on configuring the clear text port for an SSL based service, see Configuring Advanced SSL Settings.

For details on creating a certificate-key pair and binding a certificate-key pair to a service, see Adding a Certificate / key Pair.

Example

Enable SSL offloading and load balancing.

Create an SSL based service, Service-SSL-1 with the IP address 10.102.20.30 using port 443 and configure its clear text port.

Next, create a certificate-key pair, CertKey-1 and bind it to the SSL service.

Table 1. Entities in the Service-based Transparent SSL Acceleration

Entity	Name	Value
SSL Service	Service-SSL-1	102.20.30
Certificate - Key Pair	Certkey-1	

Virtual Server-based Acceleration with a Wildcard IP Address (*:443)

You can use an SSL virtual server in the wildcard IP address mode if when you want to enable SSL acceleration for multiple servers that host the secure content of a Web site. In this mode, a single-digital certificate is enough for the entire secure Web site, instead of one certificate per virtual server. This results in significant cost savings on SSL certificates and renewals. The wildcard IP address mode also enables centralized certificate management.

To configure global transparent SSL acceleration on the NetScaler appliance, create a *:443 virtual server, which is a virtual server that accepts any IP address associated with port 443. Then, bind a valid certificate to this virtual server, and also bind all services to which the virtual server is to transfer. Such a virtual server can use the SSL protocol for HTTP-based data or the SSL_TCP protocol for non-HTTP-based data.

To configure virtual server-based acceleration with a wildcard IP address

1. Enable SSL, as described in Enabling SSL Processing.
2. Enable load balancing, as described in Load Balancing.
3. Add an SSL based virtual server (see Configuring an SSL-Based Virtual Server for the basic settings), and set the **clearTextPort** parameter (described in Configuring Advanced SSL Settings).
4. Add a certificate-key pair, as described in Adding a Certificate-Key Pair.

   Note: The wildcard server will automatically learn the servers configured on the NetScaler, so you do not need to configure services for a wildcard virtual server.

Example

After enabling SSL offloading and load balancing, create an SSL based wildcard virtual server with IP address set to * and port number 443, and configure its clear text port (optional).

If you specify the clear text port, decrypted data will be sent to the backend server on that particular port. Otherwise, encrypted data will be sent to port 443.

Next, create an SSL certificate key pair, CertKey-1 and bind it to the SSL virtual server.

Table 2. Entities in the Virtual Server-based Acceleration with a Wildcard IP Address Example

Entity	Name	IP Address	Port
SSL Based Virtual Server	Vserver-SSL-Wildcard	*	443
Certificate - Key Pair	Certkey-1		

SSL VIP-based Transparent Access with End-To-End Encryption

You can use an SSL virtual server for transparent access with end-to-end encryption if you have no clear text port specified. In such a configuration, the NetScaler terminates and offloads all SSL processing, initiates a secure SSL session, and sends the encrypted data, instead of clear text data, to the web servers on the port that is configured on the wildcard virtual server.

Note: In this case, the SSL acceleration feature runs at the back-end, using the default configuration, with all 34 ciphers available.

To configure SSL VIP based transparent access with end-to-end encryption, Follow instructions for Configuring a Virtual Server-based Acceleration with a Wildcard IP Address (*:443), but do not configure a clear text port on the virtual server.

Configuring SSL Acceleration with HTTP on the Front End and SSL on the Back End

In certain deployments, you might be concerned about network vulnerabilities between the NetScaler appliance and the backend servers, or you might need complete end-to-end security and interaction with certain devices that can communicate only in clear text (for example, caching devices).

In such cases, you can set up an HTTP virtual server that receives data from clients that connect to it at the front end and hands the data off to a secure service, which securely transfers the data to the web server.

To implement this type of configuration, you configure an HTTP virtual server on the NetScaler and bind SSL based services to the virtual server. The NetScaler receives HTTP requests from the client on the configured HTTP virtual server, encrypts the data, and sends the encrypted data to the web servers in a secure SSL session.

To configure SSL acceleration with HTTP on the front-end and SSL on the back-end, first enable the load balancing and SSL features on the NetScaler. Then, add SSL based services that represent secure servers to which the NetScaler appliance will send encrypted data. Finally, add an HTTP based virtual server and bind the SSL services to this virtual server.

For details about SSL, see Enabling SSL Processing. For details about load balancing, see Load Balancing.

For details on adding an SSL based services, see Configuring Services.

For details on adding an HTTP-based virtual server, see Creating a Virtual Server.

For details on binding services to a virtual server, see Binding Services to the SSL-Based Virtual Server.

Example

Enable load balancing and SSL acceleration on the NetScaler.

After enabling load balancing and SSL acceleration, create two SSL based services, Service-SSL-1 and Service-SSL-2, with IP addreses 10.102.20.30 and 10.102.20.31, and both using port 443.

Then create an HTTP based virtual server, Vserver-HTTP-1, with an IP address of 10.102.10.20.

Bind the SSL services to the virtual server to complete the configuration.

Table 1. Entities in the SSL Acceleration with HTTP on the Front End and SSL on the Back End Example

Entity	Name	Value
SSL Service	Service-SSL-1	10.102.20.30
	Service-SSL-2	10.102.20.31
HTTP Based Virtual Server	Vserver-HTTP-1	10.102.10.20

SSL Offloading with Other TCP Protocols

In addition to the secure HTTP (HTTPS) protocol, NetScaler appliances support SSL acceleration for other TCP-based secure protocols. However, only simple requests and response-based TCP application protocols are supported. Applications such as FTPS, that insert the server's IP address and port information in their payloads, are not currently supported.

Note: The STARTTLS feature for SMTP is currently not supported.

The NetScaler supports SSL acceleration for Other TCP protocols with and without end-to-end encryption.

To configure SSL offloading with Other TCP protocols, create a virtual server of type SSL_TCP, bind a certificate-key pair and TCP based services to the virtual server, and configure SSL actions and policies based on the type of traffic expected and the acceleration to be provided.

Follow the instructions in Configuring SSL Offloading, but create an SSL_TCP virtual server instead of an SSL virtual server, and configure TCP services instead of HTTP services.

SSL_TCP Based Offloading with End-to-End Encryption

To configure SSL_TCP-based offloading with end-to-end encryption, both the virtual server that intercepts secure traffic and the services that it forwards the traffic to must be of type SSL_TCP.

Configure SSL_TCP-based offloading as described in Configuring SSL Offloading with End-to-End Encryption, but create an SSL_TCP virtual server instead of an SSL virtual server.

Backend Encryption for TCP Based Data

Some deployments might require the NetScaler appliance to encrypt TCP data received as clear text and send the data securely to the back end servers.

To provide SSL acceleration with back-end encryption for clear text TCP traffic arriving from the client, create a TCP based virtual server and bind it to SSL_TCP based services.

To configure end-to-end encryption for TCP-based data, follow the procedure described in Configuring the SSL feature with HTTP on the Front-End and SSL on the Back-End, but create a TCP virtual server instead of an HTTP virtual server.

Configuring SSL Bridging

An SSL bridge configured on the NetScaler appliance enables the appliance to bridge all secure traffic directly to the web server. The appliance does not offload or accelerate the bridged traffic. The Web server must handle all SSL-related processing. Also, features such as content switching, SureConnect, and cache redirection do not work, because the traffic passing through the NetScaler is encrypted.

Because the NetScaler does not carry out any SSL processing in an SSL bridging setup, there is no need for SSL certificates.

Citrix recommends that you use this configuration only if an acceleration unit (for example, a PCI-based SSL accelerator card) is installed in the web server to handle the SSL processing overhead.

Before you configure SSL bridging, first enable load balancing on the NetScaler. Then, create SSL_Bridge services and bind them to an SSL_Bridge virtual server. Configure the Load Balancing feature to maintain server persistency for secure requests.

For details about load balancing, see Load Balancing.

For details on adding an SSL based services, see Configuring Services.

For details on adding an SSL_Bridge virtual server, see Adding an SSL based Virtual Server.

For details on binding services to a virtual server, see Binding Services to the SSL-Based Virtual Server.

For details on configuring the load balancing feature to maintain server persistency, see Load Balancing.

Example

After enabling load balancing, create two SSL_Bridge services, Service-SSL_Bridge-1 and Service-SSL_Bridge-2, with IP addresses 192.168.1.100 and 192.168.1.101, respectively.

Then, create an SSL_Bridge virtual server, Vserver-SSL_Bridge-1, with IP address 192.168.1.10

Bind the SSL_Bridge services to the virtual server to complete the configuration.

Table 1. Entities in the SSL Bridging Example

Entity	Name	Value
SSL _Bridge Service	Service-SSL_Bridge-1	192.168.1.100
	Service-SSL_Bridge-2	192.168.1.101

SSL Based Virtual Server	Vserver-SSL_Bridge-1	192.168.1.10

Configuring the SSL Feature for Commonly Used Deployment Scenarios

Some of the most commonly deployed NetScaler SSL configurations are for load balancing secure data, applying content switching to secure data, and monitoring secure data:

- Configuring an SSL Virtual Server for Load Balancing
- Configuring a Secure Content Switching Server
- Configuring SSL Monitoring when Client Authentication is Enabled on the Backend Service

Configuring an SSL Virtual Server for Load Balancing

A virtual server configured to load balance incoming secure data first decrypts the data and then selects a web server as determined by the configured load balancing policies. The NetScaler appliance then sends the decrypted data to the selected server, using a mapped IP address as the source IP address.

To configure load balancing on the NetScaler, you must first create an SSL-based load balancing virtual server and two or more HTTP-based services. You then bind the services and an SSL certificate to the virtual server. If no load balancing policy or method is configured, the default, LEASTCONNECTION, method is used.

Example

```
> add service ssl1 10.102.29.252 HTTP 80
> add service ssl2 10.102.29.253 HTTP 80
> add lb vserver vssl  SSL 10.102.29.133 443
> bind lb vserver vssl ssl1
> bind lb vserver vssl ssl2
> add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -password ssl
> bind ssl vserver vssl certkeyName sslckey
> show ssl vserver vssl

Advanced SSL configuration for VServer vssl:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 0
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    CertKey Name: sslckey   Server Certificate

1)    Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
 Done
```

Configuring a Secure Content Switching Server

An SSL-based content switching virtual server first decrypts the secure data and then redirects the data to appropriately configured servers as determined by the type of content and the configured content switching policies. The packets sent to the server have a mapped IP address as the source IP address.

The following example shows the steps to configure two address-based virtual servers to perform load balancing on the HTTP services. One virtual server, Vserver-LB-HTML, load balances the dynamic content (cgi, asp), and the other, Vserver-LB-Image, load balances the static content (gif, jpeg). The load-balancing method used is the default, LEASTCONNECTION. A content-switching SSL virtual server, Vserver-CS-SSL, is then configured to perform SSL acceleration and switching of HTTPS requests on the basis of configured content-switching policies.

Example

```
> enable ns feature lb cs ssl
> add lb vserver Vserver-LB-HTML http 10.1.1.2 80
> add lb vserver Vserver-LB-Image http 10.1.1.3 80
> add service s1 10.1.1.4 http 80
> add service s2 10.1.1.5 http 80
> add service s3 10.1.1.6 http 80
> add service s4 10.1.1.7 http 80
> bind lb vserver Vserver-LB-HTML s1
> bind lb vserver Vserver-LB-HTML s2
> bind lb vserver Vserver-LB-Image s3
> bind lb vserver Vserver-LB-Image s4
> add cs vserver Vserver-CS-SSL ssl 10.1.1.1 443
> add cs policy pol1 -url "*.cgi"
> add cs policy pol2 -url "*.asp"
> add cs policy pol3 -url "*.gif"
> add cs policy pol4 -url "*.jpeg"
> bind cs vserver Vserver-CS-SSL -policyName pol1 Vserver-LB-HTML
> bind cs vserver Vserver-CS-SSL -policyName pol2 Vserver-LB-HTML
> bind cs vserver Vserver-CS-SSL -policyName pol3 Vserver-LB-Image
> bind cs vserver Vserver-CS-SSL -policyName pol4 Vserver-LB-Image
> add certkey mykey -cert /nsconfig/ssl/ns-root.cert -key /nsconfig/ssl/ns-root.key
> bind certkey Vserver-CS-SSL mykey
>
> show cs vserver Vserver-CS-SSL
      Vserver-CS-SSL (10.1.1.1:443) - SSL Type: CONTENT
      State: UP
      Last state change was at Tue Jul 13 02:11:37 2010
      Time since last state change: 0 days, 00:02:12.440
```

Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

Configuring SSL Monitoring when Client Authentication is Enabled on the Backend Service

Consider a scenario in which you need to load balance servers that require SSL client certificates to validate clients. For this deployment, you need to create an SSL service on the NetScaler appliance, add an HTTPS monitor, add a certificate-key pair, bind this certificate-key pair to the SSL service, and then bind the https monitor to this service. You can use this https monitor to perform health checks on the backend services.

To configure SSL monitoring with client certificate

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on the NetScaler appliance by using the administrator credentials.
3. Add an SSL service. At the command prompt, type:

   add service <name> <serverName> <serviceType> <port>
4. Add an https monitor. At the command prompt, type:

   add lb monitor <name> <type>
5. Add the certificate-key pair that is going to be used as the client cert for that SSL service. At the command prompt, type:

   add ssl certKey <certkeyName> -cert <string> -key <string>
6. Bind this certkey to the SSL service. At the command prompt, type:

   bind ssl service <serviceName> -certkeyName <string>
7. Bind the https monitor to the SSL service. At the command prompt, type:

   bind lb monitor <monitorName> <serviceName>

Now, when the NetScaler tries to probe the backend service on which client authentication is enabled, the backend service will request a certificate as part of the SSL handshake. When the NetScaler returns the certificate-key bound in step 6 above, the monitor probe will succeed.

Example

```
add service svc_k 10.102.145.30 SSL 443
add lb monitor sslmon HTTP -respCode 200 -httpRequest "GET /testsite/file5.html" -secure YES
add ssl certKey ctest -cert client_rsa_1024.pem -key client_rsa_1024.ky
bind ssl service svc_k -certkeyName ctest
bind lb monitor sslmon svc_k
> show service svc_k
     svc_k (10.102.145.30:443) - SSL
     State: UP
     Last state change was at Tue Jan 10 13:12:24 2012
     Time since last state change: 0 days, 00:09:37.890
     Server Name: 10.102.145.30
     Server ID : 0 Monitor Threshold : 0
     Max Conn: 0 Max Req: 0 Max Bandwidth: 0 kbits
     Use Source IP: NO
     Client Keepalive(CKA): NO
     Access Down Service: NO
     TCP Buffering(TCPB): NO
     HTTP Compression(CMP): NO
     Idle timeout: Client: 180 sec Server: 360 sec
     Client IP: DISABLED
     Cacheable: NO
     SC: OFF
     SP: OFF
     Down state flush: ENABLED
     Appflow logging: ENABLED

1) Monitor Name: sslmon
     State: UP Weight: 1
     Probes: 1318 Failed [Total: 738 Current: 0]
     Last response: Success - HTTP response code 200 received.
     Response Time: 0.799 millisec
  Done
>
> show ssl service svc_k
          Advanced SSL configuration for Back-end SSL Service svc_k:
     DH: DISABLED
     Ephemeral RSA: DISABLED
     Session Reuse: ENABLED Timeout: 300 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     Server Auth: DISABLED
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: DISABLED
     SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
1) CertKey Name: ctest Client Certificate

1) Cipher Name: ALL
  Description: Predefined Cipher Alias
Done
```

Ciphers Supported by the NetScaler Appliance

Your NetScaler appliance ships with a predefined set of cipher groups. Table 1 lists the ciphers that are part of the DEFAULT cipher group and are therefore bound by default to an SSL virtual server. You cannot unbind the DEFAULT cipher group from an SSL virtual server. Table 2 lists the other ciphers currently supported by the NetScaler appliance. To use ciphers that are not part of the DEFAULT cipher group, you have to explicitly bind them to an SSL virtual server. You can also create a user-defined cipher group to bind to the SSL virtual server. For more information about creating a user-defined cipher group, see Configuring User-Defined Cipher Groups on the NetScaler Appliance.

Table 1. Ciphers That the NetScaler Appliance Supports by Default

Cipher Suite	Protocol	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm
SSL3-RC4-MD5	SSLv3	RSA	RSA	RC4(128)	MD5
SSL3-RC4-SHA	SSLv3	RSA	RSA	RC4(128)	SHA1
SSL3-DES-CBC3-SHA	SSLv3	RSA	RSA	3DES(168)	SHA1
TLS1-AES-256-CBC-SHA	TLSv1	RSA	RSA	AES(256)	SHA1
TLS1-AES-128-CBC-SHA	TLSv1	RSA	RSA	AES(128)	SHA1
SSL3-EDH-DSS-DES-CBC3-SHA	SSLv3	DH	DSS	3DES(168)	SHA1
TLS1-DHE-DSS-RC4-SHA	TLSv1	DH	DSS	RC4(128)	SHA1
TLS1-DHE-DSS-AES-256-CBC-SHA	TLSv1	DH	DSS	AES(256)	SHA1
TLS1-DHE-DSS-AES-128-CBC-SHA	TLSv1	DH	DSS	AES(128)	SHA1
SSL3-EDH-RSA-DES-CBC3-SHA	SSLv3	DH	RSA	3DES(168)	SHA1
TLS1-DHE-RSA-AES-256-CBC-SHA	TLSv1	DH	RSA	AES(256)	SHA1
TLS1-DHE-RSA-AES-128-CBC-SHA	TLSv1	DH	RSA	AES(128)	SHA1

Table 2. Additional Ciphers Supported by the NetScaler Appliance

Cipher Suite	Protocol	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm
SSL3-DES-CBC-SHA	SSLv3	RSA	RSA	DES(56)	SHA1
TLS1-EXP1024-RC4-SHA	TLSv1	RSA(1024)	RSA	RC4(56)	SHA1 Export

SSL3-EXP-RC4-MD5	SSLv3	RSA(512)	RSA	RC4(40)	MD5 Export
SSL3-EXP-DES-CBC-SHA	SSLv3	RSA(512)	RSA	DES(40)	SHA1 Export
SSL3-EXP-RC2-CBC-MD5	SSLv3	RSA(512)	RSA	RC2(40)	MD5 Export
SSL2-RC4-MD5	SSLv2	RSA	RSA	RC4(128)	MD5
SSL2-DES-CBC3-MD5	SSLv2	RSA	RSA	3DES(168)	MD5
SSL2-RC2-CBC-MD5	SSLv2	RSA	RSA	RC2(128)	MD5
SSL2-DES-CBC-MD5	SSLv2	RSA	RSA	DES(56)	MD5
SSL2-RC4-64-MD5	SSLv2	RSA	RSA	RC4(64)	MD5
SSL2-EXP-RC4-MD5	SSLv2	RSA(512)	RSA	RC4(40)	MD5 Export
SSL3-EDH-DSS-DES-CBC-SHA	SSLv3	DH	DSS	DES(56)	SHA1
TLS1-EXP1024-DHE-DSS-DES-CBC-SHA	TLSv1	DH(1024)	DSS	DES(56)	SHA1 Export
TLS1-EXP1024-DHE-DSS-RC4- SHA	TLSv1	DH(1024)	DSS	RC4(56)	SHA1 Export
SSL3-EXP-EDH-DSS-DES-CBC-SHA	SSLv3	DH(512)	DSS	DES(40)	SHA1 Export
SSL3-EDH-RSA-DES-CBC-SHA	SSLv3	DH	RSA	DES(56)	SHA1
SSL3-EXP-EDH-RSA-DES-CBC-SHA	SSLv3	DH(512)	RSA	DES(40)	DES(40)
TLS1-EXP1024-RC4-MD5	TLSv1	RSA(1024)	RSA	RC4(56)	MD5 Export
TLS1-EXP1024-RC2-CBC-MD5	TLSv1	RSA(1024)	RSA	RC2(56)	MD5 Export
SSL2-EXP-RC2-CBC-MD5	SSLv2	RSA(512)	RSA	RC2(40)	MD5 Export
SSL3-ADH-RC4-MD5	SSLv3	DH	None	RC4(128)	MD5
SSL3-ADH-DES-CBC-SHA	SSLv3	DH	None	DES(56)	SHA1
SSL3-ADH-DES-CBC3-SHA	SSLv3	DH	None	3DES(168)	SHA1
TLS1-ADH-AES-128-CBC-SHA	TLSv1	DH	None	AES(128)	SHA1
TLS1-ADH-AES-256-CBC-SHA	TLSv1	DH	None	AES(256)	SHA1
SSL3-EXP-ADH-RC4-MD5	SSLv3	DH(512)	None	RC4(40)	MD5 Export
SSL3-EXP-ADH-DES-CBC-SHA	SSLv3	DH(512)	None	DES(40)	SHA1 Export

FIPS

The Federal Information Processing Standard (FIPS), issued by the US National Institute of Standards and Technologies, specifies the security requirements for a cryptographic module used in a security system. The NetScaler FIPS appliance complies with the second version of this standard, FIPS-140-2.

Note: Henceforth, all references to FIPS imply FIPS-140-2.

The FIPS appliance is equipped with a tamper-proof (tamper-evident) cryptographic module—a Cavium CN1120-NFB card on the 9950 FIPS and 9010 FIPS and a Cavium CN1120-NFBE3-2.0-G on the MPX 9700/10500/12500/15500 FIPS appliances—designed to comply with the FIPS 140-2 Level-2 and Level-3 specifications. The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (nsroot) can perform operations on the keys stored inside the HSM.

The following table summarizes the differences between standard NetScaler and NetScaler FIPS appliances.

Setting	NetScaler appliance	NetScaler FIPS appliance
Key storage	On the hard disk	On the FIPS card
Cipher support	All ciphers	FIPS approved ciphers
Accessing keys	From the hard disk	Not accessible

Note: The only non-FIPS cipher supported on the NetScaler 9010 and 9950 FIPS appliances is SSL3-RC4-SHA.This cipher is not supported on the MPX 9700/10500/12500/15500 FIPS appliances. Only the Citrix NetScaler MPX 9700/10500/12500/15500 FIPS appliances support the NetScaler 9.3 nCore software.

Configuring a FIPS appliance involves configuring the HSM immediately after completing the generic configuration process. You then create or import a FIPS key. After creating a FIPS key, you should export it for backup. You might also need to export a FIPS key so that you can import it to another appliance. For example, configuring FIPS appliances in a high availability (HA) setup requires transferring the FIPS key from the primary node to the secondary node immediately after completing the standard HA setup.

You can upgrade the firmware version on the FIPS card from version 4.6.0 to 4.6.1, and you can reset an HSM that has been locked to prevent unauthorized logon. Only FIPS approved ciphers are supported on a NetScaler FIPS appliance.

Configuring the HSM

Before you can configure the HSM of your NetScaler FIPS appliance, you must complete the initial hardware configuration. For more information, see Initial Configuration.

Configuring the HSM of your NetScaler FIPS appliance erases all existing data on the HSM. To configure the HSM, you must be logged on to the appliance as the superuser (nsroot account). The HSM is preconfigured with default values for the Security Officer (SO) password and User password, which you use to configure the HSM or reset a locked HSM.

The procedure for configuring the HSM on MPX 9700/10500/12500/15500 FIPS appliances is different from the procedure for 9010 and 9950 FIPS appliances. On the MPX 9700/10500/12500/15500 FIPS appliances, you need to check the status of the FIPS card and reset it before initializing the FIPS card. You also need to restart the appliance after executing the reset fips command, and after executing set fips command, so that the command takes effect before you execute the next command. Neither the preliminary steps nor the restarts are required for configuring the HSM on a 9010 FIPS or 9950 FIPS appliance. Be sure to use the appropriate procedure for your appliance. Also, the two groups of appliances have different default values for the Security Officer (SO) and user passwords.

Although the FIPS appliance can be used with the default password values, you should modify them before using it. The HSM can be configured only when you log on to the appliance as the superuser and specify the SO and User passwords.

Important: Due to security constraints, the appliance does not provide a means for retrieving the SO password. Store a copy of the password safely. Should you need to reinitialize the HSM, you will need to specify this password as the old SO password.

After upgrading, verify that the /nsconfig/fips directory has been successfully created on the appliance.

To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the NetScaler command line

After logging on to the appliance as the superuser and completing the initial configuration, at the NetScaler command prompt, type the following commands to configure the HSM and verify the configuration:

1. show fips
2. reset fips
3. reboot -warm

4. set ssl fips -**initHSM Level-2** <newSOpassword> <oldSOpassword> <userPassword> [-**hsmLabel** <string>]

5. saveconfig

6. reboot -warm

7. show fips

Example

```
show fips
FIPS Card is not configured
Done
reset fips
reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:y
set ssl fips -initHSM Level-2 sopin12345 so12345 user123 -hsmLabel cavium
This command will erase all data on the FIPS card. You must save the configuration
(saveconfig) after executing this command.

Do you want to continue?(Y/N)y
Done
saveconfig
reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:y
show fips
       FIPS HSM Info:
HSM Label            : cavium
Initialization       : FIPS-140-2 Level-2
HSM Serial Number    : 2.1G1008-IC000007
HSM State            : 2
Firmware Version     : 1.1
Firmware Release Date  : Jun04,2010

Max FIPS Key Memory    : 3996
Free FIPS Key Memory   : 3994
Total SRAM Memory      : 467348
Free SRAM Memory       : 62552
Total Crypto Cores     : 3
Enabled Crypto Cores   : 3
Done
```

To configure the HSM on a 9010 FIPS or 9950 FIPS appliance by using the NetScaler command line

After logging on to the appliance as the superuser and completing the initial configuration, at the NetScaler command prompt, type the following commands to configure the HSM and verify the configuration:

- set ssl fips -**initHSM Level-2** <new SO password> <old SO password> <user password> [-**hsmLabel** <string>]
- show ssl fips

Example

set ssl fips -initHSM Level-2 fipssso123 sopin123 userpin123 -hsmLabel FIPS-140-2

This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command.

Do you want to continue? (Y/N) y

show ssl fips

FIPS HSM Info:
HSM Label : FIPS-140-2
Initialization : FIPS-140-2 Level-2
HSM Serial Number : 8007376
Firmware Version : 4.6.1
Total Flash Memory : 14286412
Free Flash Memory : 14281588
Total SRAM Memory : 17036680
Free SRAM Memory : 17035240

Parameters for configuring the HSM

initHSM

The FIPS initialization level. The appliance currently supports Level-2 (FIPS 140-2 Level-2). Possible value: Level 2.

hsmLabel

The label to identify the Hardware Security Module (HSM). Maximum Length: 31.

newSOpassword

The security officer password that will be in effect after you have configured the HSM. Maximum length on 9010 and 9950 FIPS appliances: 31 characters. Maximum length on MPX 9700/10500/12500/15500 FIPS appliances: 14 characters.

oldSOpassword

The old security office password. Default on MPX 9700/10500/12500/15500 FIPS appliances: so12345. Default on 9010 and 9950 FIPS appliances: sopin123.

userPassword

The user password. Default on MPX 9700/10500/12500/15500 FIPS appliances: user123. Default on 9010 and 9950 FIPS appliances: userpin123. Maximum length on 9010 and 9950 FIPS appliances: 31 characters. Maximum length on MPX 9700/10500/12500/15500 FIPS appliances: 14 characters.

To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**. In the details pane, verify that the message "FIPS card is not configured" appears.
2. In the details pane, on the **FIPS Info**tab, click **Reset FIPS**.
3. In the navigation pane, click **System**.
4. In the details pane, click **Reboot**.
5. In the details pane, on the **FIPS Info** tab, click **Initialize HSM**.
6. In the **Initialize HSM** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring the HSM” as shown:
   - **Security Officer (SO) Password***—new SO password
   - **Old SO Password***—old SO password
   - **User Password***—user password
   - **Level**—initHSM (Currently set to Level2 and cannot be changed)
   - **HSM Label**—hsmLabel

   *A required parameter
7. Click **OK**.
8. In the details pane, click **Save**.
9. In the navigation pane, click **System**.
10. In the details pane, click **Reboot**.
11. Under FIPS HSM Info, verify that the information displayed for the FIPS HSM that you just configured is correct.

To configure the HSM on a 9010 FIPS or 9950 FIPS appliance by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Info** tab, click **Initialize HSM**.
3. In the **Initialize HSM** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring the HSM" as shown:
   - **Security Officer (SO) Password***—newSOpassword
   - **Old SO Password***—oldSOpassword
   - **User Password***—userpassword
   - **Level**—initHSM (Currently set to Level 2 and cannot be changed)
   - **HSM Label**—hsmLabel

   *A required parameter
4. Click **OK**.
5. Under FIPS HSM Info, verify that the information displayed for the FIPS HSM that you just configured is correct.

Important: After the HSM is initialized, the current configuration on the appliance needs to be saved. If this is not done, the card will not function after the appliance is restarted, and three unsuccessful attempts to change the SO password will cause the card to be locked. To reset a locked HSM, see Resetting a Locked HSM.

Note: Citrix recommends that you store the SO password in a secure location. You will need to specify this password as the old SO password to re-initialize the HSM.

Important: If you want to upgrade to the latest software release, see the *Citrix NetScaler Migration Guide* at http://support.citrix.com/article/CTX128675. In the installation steps, use the ./installns -F command to install FIPS.

Configuring the HSM

Before you can configure the HSM of your NetScaler FIPS appliance, you must complete the initial hardware configuration. For more information, see Initial Configuration.

Configuring the HSM of your NetScaler FIPS appliance erases all existing data on the HSM. To configure the HSM, you must be logged on to the appliance as the superuser (nsroot account). The HSM is preconfigured with default values for the Security Officer (SO) password and User password, which you use to configure the HSM or reset a locked HSM.

The procedure for configuring the HSM on MPX 9700/10500/12500/15500 FIPS appliances is different from the procedure for 9010 and 9950 FIPS appliances. On the MPX 9700/10500/12500/15500 FIPS appliances, you need to check the status of the FIPS card and reset it before initializing the FIPS card. You also need to restart the appliance after executing the reset fips command, and after executing set fips command, so that the command takes effect before you execute the next command. Neither the preliminary steps nor the restarts are required for configuring the HSM on a 9010 FIPS or 9950 FIPS appliance. Be sure to use the appropriate procedure for your appliance. Also, the two groups of appliances have different default values for the Security Officer (SO) and user passwords.

Although the FIPS appliance can be used with the default password values, you should modify them before using it. The HSM can be configured only when you log on to the appliance as the superuser and specify the SO and User passwords.

Important: Due to security constraints, the appliance does not provide a means for retrieving the SO password. Store a copy of the password safely. Should you need to reinitialize the HSM, you will need to specify this password as the old SO password.

After upgrading, verify that the /nsconfig/fips directory has been successfully created on the appliance.

To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the NetScaler command line

After logging on to the appliance as the superuser and completing the initial configuration, at the NetScaler command prompt, type the following commands to configure the HSM and verify the configuration:

1. show fips
2. reset fips
3. reboot -warm

4. set ssl fips -**initHSM Level-2** <newSOpassword> <oldSOpassword> <userPassword> [-**hsmLabel** <string>]

5. saveconfig

6. reboot -warm

7. show fips

Example

```
show fips
FIPS Card is not configured
Done
reset fips
reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:y
set ssl fips -initHSM Level-2 sopin12345 so12345 user123 -hsmLabel cavium
This command will erase all data on the FIPS card. You must save the configuration
(saveconfig) after executing this command.

Do you want to continue?(Y/N)y
Done
saveconfig
reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:y
show fips
       FIPS HSM Info:
HSM Label            : cavium
Initialization       : FIPS-140-2 Level-2
HSM Serial Number    : 2.1G1008-IC000007
HSM State            : 2
Firmware Version     : 1.1
Firmware Release Date  : Jun04,2010

Max FIPS Key Memory    : 3996
Free FIPS Key Memory   : 3994
Total SRAM Memory      : 467348
Free SRAM Memory       : 62552
Total Crypto Cores     : 3
Enabled Crypto Cores   : 3
Done
```

To configure the HSM on a 9010 FIPS or 9950 FIPS appliance by using the NetScaler command line

After logging on to the appliance as the superuser and completing the initial configuration, at the NetScaler command prompt, type the following commands to configure the HSM and verify the configuration:

- set ssl fips -**initHSM Level-2** <new SO password> <old SO password> <user password> [-**hsmLabel** <string>]
- show ssl fips

Example

set ssl fips -initHSM Level-2 fipssso123 sopin123 userpin123 -hsmLabel FIPS-140-2

This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command.

Do you want to continue? (Y/N) y

show ssl fips

FIPS HSM Info:
HSM Label : FIPS-140-2
Initialization : FIPS-140-2 Level-2
HSM Serial Number : 8007376
Firmware Version : 4.6.1
Total Flash Memory : 14286412
Free Flash Memory : 14281588
Total SRAM Memory : 17036680
Free SRAM Memory : 17035240

Parameters for configuring the HSM

initHSM

The FIPS initialization level. The appliance currently supports Level-2 (FIPS 140-2 Level-2). Possible value: Level 2.

hsmLabel

The label to identify the Hardware Security Module (HSM). Maximum Length: 31.

newSOpassword

The security officer password that will be in effect after you have configured the HSM. Maximum length on 9010 and 9950 FIPS appliances: 31 characters. Maximum length on MPX 9700/10500/12500/15500 FIPS appliances: 14 characters.

oldSOpassword

The old security office password. Default on MPX 9700/10500/12500/15500 FIPS appliances: so12345. Default on 9010 and 9950 FIPS appliances: sopin123.

userPassword

The user password. Default on MPX 9700/10500/12500/15500 FIPS appliances: user123. Default on 9010 and 9950 FIPS appliances: userpin123. Maximum length on 9010 and 9950 FIPS appliances: 31 characters. Maximum length on MPX 9700/10500/12500/15500 FIPS appliances: 14 characters.

To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**. In the details pane, verify that the message "FIPS card is not configured" appears.
2. In the details pane, on the **FIPS Info**tab, click **Reset FIPS**.
3. In the navigation pane, click **System**.
4. In the details pane, click **Reboot**.
5. In the details pane, on the **FIPS Info** tab, click **Initialize HSM**.
6. In the **Initialize HSM** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for configuring the HSM” as shown:
   - **Security Officer (SO) Password***—new SO password
   - **Old SO Password***—old SO password
   - **User Password***—user password
   - **Level**—initHSM (Currently set to Level2 and cannot be changed)
   - **HSM Label**—hsmLabel

   *A required parameter
7. Click **OK**.
8. In the details pane, click **Save**.
9. In the navigation pane, click **System**.
10. In the details pane, click **Reboot**.
11. Under FIPS HSM Info, verify that the information displayed for the FIPS HSM that you just configured is correct.

To configure the HSM on a 9010 FIPS or 9950 FIPS appliance by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Info** tab, click **Initialize HSM**.
3. In the **Initialize HSM** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring the HSM" as shown:
   - **Security Officer (SO) Password***—newSOpassword
   - **Old SO Password***—oldSOpassword
   - **User Password***—userpassword
   - **Level**—initHSM (Currently set to Level 2 and cannot be changed)
   - **HSM Label**—hsmLabel

   *A required parameter
4. Click **OK**.
5. Under FIPS HSM Info, verify that the information displayed for the FIPS HSM that you just configured is correct.

Important: After the HSM is initialized, the current configuration on the appliance needs to be saved. If this is not done, the card will not function after the appliance is restarted, and three unsuccessful attempts to change the SO password will cause the card to be locked. To reset a locked HSM, see Resetting a Locked HSM.

Note: Citrix recommends that you store the SO password in a secure location. You will need to specify this password as the old SO password to re-initialize the HSM.

Important: If you want to upgrade to the latest software release, see the *Citrix NetScaler Migration Guide* at http://support.citrix.com/article/CTX128675. In the installation steps, use the ./installns -F command to install FIPS.

Creating and Transferring FIPS Keys

After configuring the HSM of your FIPS appliance, you are ready to create a FIPS key. The FIPS key is created in the appliance's HSM. You can then export the FIPS key to the appliance's CompactFlash card as a secured backup. Exporting the key also enables you to transfer it by copying it to the /flash of another appliance and then importing it into the HSM of that appliance.

Instead of creating a FIPS key, you can import an existing FIPS key or import an external key as a FIPS key.

Note: If you are planning an HA setup, make sure that the FIPS appliances are configured in an HA setup before creating a FIPS key.

Creating a FIPS Key

Before creating a FIPS key, make sure that the HSM is configured.

To create a FIPS key by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a FIPS key and verify the settings:

- create ssl fipsKey <fipsKeyName> -**modulus** <positive_integer> [-**exponent** (3 | F4)]
- show ssl fipsKey [<fipsKeyName>]

Example

```
create fipskey Key-FIPS-1 -modulus 2048 -exponent 3
show ssl fipsKey Key-FIPS-1
FIPS Key Name: Key-FIPS-1 Modulus: 2048   Public Exponent: 3 (Hex: 0x3)
```

Parameters for Creating a FIPS Key

fipsKeyName

The object name for the FIPS key. Maximum Length: 31.

modulus

The modulus of the key to be created. The modulus value should be a multiple of 64. Possible values on 9010 and 9950 FIPS appliances: 512, 1024, 2048. Possible values on MPX 9700/10500/12500/15500 FIPS appliances: 1024, 2048.

exponent

The exponent value for the key to be created. Possible values: 3 (Hex: 0x3), F4 (Hex: 0x00001). Default: 3.

To create a FIPS key by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Keys** tab, click **Add**.
3. In the **Create FIPS Key** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for Creating a FIPS Key” as shown:
   - FIPS Key Name*—fipsKeyName
   - Modulus*—modulus
   - Exponent*—exponent

   *A required parameter
4. Click **Create**, and then click **Close**.
5. On the **FIPS Keys** tab, verify that the settings displayed for the FIPS key that you just created are correct.

Exporting a FIPS Key

Citrix recommends that you create a backup of any key created in the FIPS HSM. If a key in the HSM is deleted, there is no way to create the same key again, and all the certificates associated with it are rendered useless.

In addition to exporting a key as a backup, you might need to export a key for transfer to another appliance.

The following procedure provides instructions on exporting a FIPS key to the /nsconfig/ssl folder on the appliance's CompactFlash and securing the exported key by using a strong asymmetric key encryption method.

To export a FIPS key by using the NetScaler command line

At the NetScaler command prompt, type:

export ssl fipsKey <fipsKeyName> -**key** <string>

Example

export fipskey Key-FIPS-1 -key Key-FIPS-1.key

Parameters for exporting a FIPS key

fipsKeyName

The name of the FIPS key to be exported. Maximum Length: 31.

key

The path and file name in which to store the exported key. Maximum Length: 63. Default path: /nsconfig/ssl/.

To export a FIPS key by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Keys** tab, click **Export**.
3. In the **Export FIPS key to a file** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for exporting a FIPS key” as shown:
   - FIPS Key Name*—fipsKeyName
   - File Name*—key (To put the file in a location other than the default, you can either specify the complete path or click the Browse button and navigate to a location.)

   *A required parameter
4. Click **Export**, and then click **Close**.

Importing an Existing FIPS Key

To use an existing FIPS key with your FIPS appliance, you need to transfer the FIPS key from the hard disk of the appliance into its HSM.

In an HA setup, the FIPS key in the primary and secondary nodes should be the same. After exporting the key to the CompactFlash of the primary node, you copy it to the secondary node's CompactFlash and then import it to secondary node's HSM.

Note: To avoid errors when importing a FIPS key, make sure that the name of the key imported is the same as the original key name when it was created.

To import a FIPS key on the 9700/10500/12500/15500 FIPS appliances by using the NetScaler command line

At the NetScaler command prompt, type the following commands to import a FIPS key and verify the settings:

- import ssl fipsKey <fipsKeyName> -**key** <string> -**inform SIM** -**exponent** (F4 | 3)
- show ssl fipskey <fipsKeyName>

Example

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
show ssl fipskey key-FIPS-2
FIPS Key Name: Key-FIPS-2 Modulus: 2048   Public Exponent: F4 (Hex value 0x10001)
```

To import a FIPS key on the 9010 FIPS and 9950 FIPS by using the NetScaler command line

At the NetScaler command prompt, type the following commands to import a FIPS key and verify the settings:

- import ssl fipsKey <fipsKeyName> -**key** <string> -**inform SIM**
- show ssl fipskey <fipsKeyName>

Example

```
import fipskey Key-FIPS-1 -key Key-FIPS-1.key -inform SIM
show ssl fipskey key-FIPS-1
```

FIPS Key Name: Key-FIPS-1 Modulus: 2048 Public Exponent: 3 (Hex: 0x3)

Parameters for importing an existing FIPS key

fipsKeyName

The name of the FIPS key to be imported. Maximum Length: 31.

key

The name of the key file. By default, the file is placed in the /nsconfig/ssl/ directory. If you want to put the file in a different location, include the complete path.

inform

The input format of the key file. Possible value: SIM (Secure Information Management; used when a FIPS key is transferred from one FIPS appliance to another).

exponent

The exponent value for the FIPS key to be imported. Possible values: 3 and F4. Default: F4.

To import a FIPS key by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Keys** tab, click **Import**.
3. In the **Import as a FIPS Key** dialog box, select **FIPS key file** and specify values for the following parameters, which correspond to parameters described in “Parameters for importing an existing FIPS Key” as shown:
   - FIPS Key Name*—fipsKeyName
   - File Name*—key (To put the file in a location other than the default, you can either specify the complete path or click the Browse button and navigate to a location.)
   - Exponent*—exponent (The exponent parameter is not required on the 9010 FIPS and 9950 FIPS appliances.)

   *A required parameter
4. Click **Import**, and then click **Close**.
5. On the **FIPS Keys** tab, verify that the settings displayed for the FIPS key that you just imported are correct.

Importing External Keys

In addition to transferring FIPS keys that are created within the NetScaler appliance's HSM, you can transfer external private keys (such as those created on a standard NetScaler, Apache, or IIS) to a FIPS NetScaler. External keys are created outside the HSM, by using a tool such as OpenSSL. Before importing an external key into the HSM, copy it to the appliance's flash drive under /nsconfig/ssl. The 9010 and 9950 FIPS appliances require additional steps before importing the key.

Importing a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the NetScaler command line

On the MPX 9700/10500/12500/15500 FIPS appliances, the -**exponent** parameter in the import fipskey command is not required while importing an external key. The correct public exponent is detected automatically when the key is imported, and the value of the -**exponent** parameter is ignored.

The NetScaler FIPS appliance does not support external keys with a public exponent other than 3 or F4.

You do not need a wrap key on the MPX 9700/10500/12500/15500 FIPS appliances.

You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 10G FIPS appliance. To import the key you need to first decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>

To import a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the NetScaler command line

1. Copy the external key to the appliance's flash drive.
2. At the NetScaler command prompt, type the following commands to import the FIPS key and verify the settings:
   - import ssl fipsKey <fipsKeyName> - **key** <string> - **inform PEM**
   - show ssl fipskey<fipsKeyName>

Example

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
show ssl fipskey key-FIPS-2
FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0x10001)
```

Note: The modulus is incorrectly displayed as zero in the above example. The discrepancy does not affect SSL functionality.

Importing a FIPS key on the 9010 and 9950 appliances by using the NetScaler command line

On the 9010 and 9950 FIPS appliances, you need to generate a wrap key and then convert the external key to PKCS8 format before importing an external key to the HSM. The wrap key encrypts the external key when it is imported.

Important: Before performing the following procedure, transfer the external FIPS key to the NetScaler appliance's flash drive.

To generate a wrap key by using the NetScaler command line

At the NetScaler command prompt, type the following commands to generate a wrap key and verify the settings:

- create ssl wrapkey <wrapKeyName> -**password** <string> -**salt** <string>
- show ssl wrapkey

Example

```
create wrapkey Key-Wrap-1 -password wrapkey123 -salt wrapsalt123
show ssl wrapkey
1)      WRAP Key Name: Key-Wrap-1
```

Parameters for generating a wrap key

wrapKeyName

The object name for the wrap key. Maximum Length: 31.

password

The password string for the wrap key. Maximum Length: 31.

salt

The salt string for the wrap key. Maximum Length: 31.

To generate a wrap key by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **Wrap Keys** tab, click **Add**.
3. In the **Create Wrap Key** dialog box, specify values for the following parameters, which correspond to parameters described in “Parameters for generating a wrap key” as shown:
   - Wrap Key Name*—wrapKeyName
   - Password*— password
   - Salt*—salt

   *A required parameter
4. Click **Create**, and then click **Close**.
5. On the **Wrap Keys** tab, verify that the settings displayed for the wrap key that you just created are correct.

Converting the External Key to PKCS8 Format and Importing it

You must convert the external key to PKCS8 format before it can be imported into the HSM. At the NetScaler command line, you enter two commands to convert and import the key. In the configuration utility, you select options from a dialog box.

To convert an external key to the PKCS8 format and import it into the HSM by using the NetScaler command line

Important: Before performing the following procedure, transfer the external FIPS key to the NetScaler appliance’s flash drive.

At the NetScaler command prompt, type the following commands to convert and import a FIPS key and verify the settings:

- convert ssl pkcs8 <pkcs8File> <keyFile> [-**keyform** (DER | PEM)]
- import ssl fipsKey <fipsKeyName> -**key** <string> [-**inform DER**] [-**wrapKeyName** <string>] [-**iv** <string>]
- show ssl fipskey <fipsKeyName>

Example

convert ssl pkcs8 Key-PKCS8-1 Key-External-1.pem -keyform PEM

Enter PEM pass phrase:

Done

import fipskey Key-Pkcs8-1 -key Key-Pkcs8-1.key -inform DER -wrapKeyName Key-Wrap-1 -iv wrap123

show ssl fipskey Key-Pkcs8-1

FIPS Key Name: Key-Pkcs8-1 Modulus: 2048 Public Exponent: 3 (Hex:0x3)

Note: When you specify the keyform as PEM and the PEM key is encrypted, you are prompted for a password.

Parameters for converting an external key to PKCS8 format

pkcs8File

The name of the output file in which the PKCS8 format key file is stored. The default output path for the PKCS8 file is /nsconfig/ssl/. Maximum value: 63.

keyFile

The input key file. The default input path for the key file is /nsconfig/ssl/. Maximum value: 63.

keyform

The format of the keyFile. Possible values: DER (Distinguished Encoding Rule).

pass phrase

The password (optional) used to encrypt the external key in PEM format. The CLI prompts you for the password at run-time.

Parameters for importing an external key into the HSM

fipsKeyName

The object name for the FIPS key being imported. Maximum Length: 31 characters.

key

The path to the key file. The default input path for the key is /nsconfig/ssl/. Maximum Length: 63 characters.

inform

The input format of the key file. Possible values: SIM (Secure Information Management; used when a FIPS key is transferred from one FIPS appliance to another), DER; used when

importing a non-FIPS key to a FIPS appliance).

wrapKeyName

The object name of the wrapkey to use for importing the key. Required if the key being imported is a non-FIPS key. Maximum Length: 31 characters.

iv

The initialization vector (IV) to use for importing the key. Required if the key being imported is a non-FIPS key. Maximum Length: 7 characters.

To convert the external key into the PKCS8 format and import it into the HSM by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the **FIPS** pane, on the **FIPS Keys** tab, click **Import**.
3. In the **Import as a FIPS Key** dialog box, select **Pkcs8 file** and click **Convert**.
4. In the **Convert private key to PKCS8 format** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for converting an external key into PKCS8 format" as shown:
   - Key Name (pkcs8 format)*—pkcs8File (To select an existing output file, click the Browse button and select from the default location or navigate to a location.)
   - Private Key Path*— keyFile (To select an existing input file, click the Browse button and select from the default location or navigate to a location.)
   - Key Format—keyform
   - Password—pass phrase

   *A required parameter
5. Click **Convert**, and then click **Close**.
6. In the **Import as a FIPS Key** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for importing an external key into the HSM"
   - FIPS Key Name*—fipsKeyName
   - File Name*—key
   - Input Format*—inform
   - Wrap Key Name*—wrapKeyName
   - IV*—iv

   *A required parameter
7. Click **Import**, and then click **Close**.
8. In the **FIPS Keys** tab, verify that the settings displayed for the external key that you just converted and imported are correct.

Note: For security reasons, delete the external private key from the hard disk after you import it into the HSM.

Securely Transferring FIPS Keys between Two Appliances

Secure Inform Management (SIM) is a secure method for transferring FIPS keys between two appliances if the appliances are not configured in a high availability setup.

Note: You should use the NetScaler command line to perform this procedure.

The following procedure transfers FIPS keys from the source appliance (appliance A) to the target appliance (appliance B).

To securely transfer FIPS keys from appliance A to appliance B by using the NetScaler command line

1. **On appliance A**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance, using the administrator credentials.
3. Initialize appliance A as the source appliance. At the NetScaler command line, type:

   init ssl fipsSIMsource <certFile>
4. Copy this <certFile> file to appliance B, in the /nconfig/ssl folder.
5. **On appliance B**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
6. Log on to the appliance, using the administrator credentials.
7. Initialize appliance B as the target appliance. At the NetScaler command line, type:

   init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
8. Copy this <targetSecret> file to appliance A.
9. **On appliance A**, enable appliance A as the source appliance. At the NetScaler prompt, type:

   enable ssl fipsSIMSource <targetSecret> <sourceSecret>
10. Copy this <sourceSecret> file to appliance B.
11. **On appliance B**, enable appliance B as the target appliance. At the NetScaler prompt, type:

    enable ssl fipsSIMtarget <keyVector> <sourceSecret>
12. **On appliance A**, create a FIPS key, as described in Creating a FIPS Key, or use an existing FIPS key.
13. Export the FIPS key to the appliance's hard disk, as described in Exporting a FIPS Key.
14. Copy the FIPS key to the hard disk of the secondary appliance by using a secure file transfer utility, such as SCP.
15. **On appliance B**, import the FIPS key from the hard disk into the HSM of the appliance, as described in Importing an Existing FIPS Key.

Example

In the following example, source.cert is the certificate on the source appliance, stored in the default directory, /nsconfig/ssl. This certificate must be transferred to the same location (/nsconfig/ssl) on the target appliance. The file target.secret is created on the

target appliance and copied to the source appliance. The file source.secret is created on the source appliance and copied to the target appliance.

On the source appliance

```
init fipsSIMsource /nsconfig/ssl/source.cert
```

On the target appliance

```
init fipsSIMtarget/nsconfig/ssl/source.cert /nsconfig/ssl/target.key /nsconfig/ssl/target.secret
```

On the source appliance

```
enable fipsSIMsource /nsconfig/ssl/target.secret /nsconfig/ssl/source.secret
```

On the target appliance

```
enable fipsSIMtarget /nsconfig/ssl/target.key /nsconfig/ssl/source.secret
```

On the source appliance

```
create fipskey fips1 -modulus 1024 -exponent f4
export fipskey fips1 -key /nsconfig/ssl/fips1.key
```

Copy this key to the hard disk of the target appliance.

On the target appliance

```
import fipskey fips1 -key /nsconfig/ssl/fips1.key
```

The following diagram summarizes the transfer process

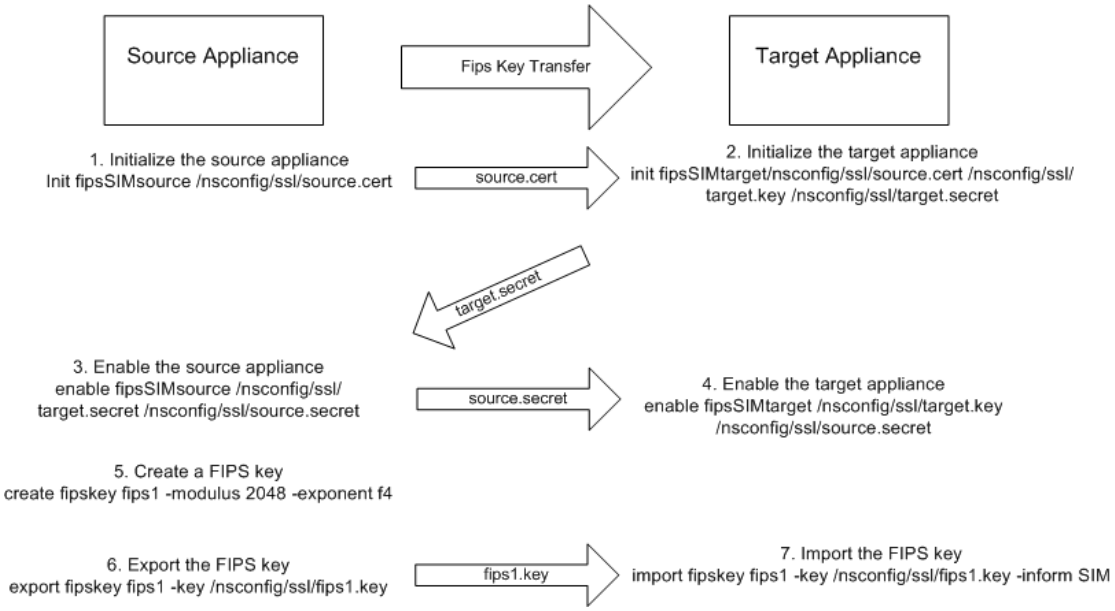


Figure 1. Transferring the FIPS Key-Summary

Configuring FIPS Appliances in a High Availability Setup

You can configure two appliances in a high availability (HA) pair as FIPS appliances. For information about configuring an HA setup, see Configuring High Availability

Note: Command line users might want to use the graphical interface (the configuration utility) to configure FIPS HA. If you use the CLI, the create fipskey command is not propagated to the secondary node. When you execute the command with the same input values for modulus size and exponent on two different FIPS appliances, the keys generated are not identical. You have to create the FIPS key on one of the nodes and then transfer it to the other node. But if you use the configuration utility to configure FIPS appliances in an HA setup, the FIPS key that you create is automatically transferred to the secondary node. The process of managing and transferring the FIPS keys is known as secure information management (SIM).

Important: On the MPX 9700/10500/12500/15500 FIPS appliances, the HA setup should be completed within six minutes. If the process takes longer than six minutes, the internal timer of the FIPS card expires and the following error message appears:

ERROR: Operation timed out or repeated, please wait for 10 mins and redo the SIM/HA configuration steps.

If this message appears, restart the appliance or wait for 10 minutes, and then repeat the HA setup procedure.

In the following procedure, appliance A is the primary node and appliance B is the secondary node.

To configure FIPS appliances in a high availability setup by using the NetScaler command line

1. **On appliance A**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance, using the administrator credentials.
3. Initialize appliance A as the source appliance. At the NetScaler command line, type:

   init ssl fipsSIMsource <certFile>
4. Copy this <certFile> file to appliance B, in the /nconfig/ssl folder.
5. **On appliance B**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
6. Log on to the appliance, using the administrator credentials.
7. Initialize appliance B as the target appliance. At the NetScaler command line, type:

   init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
8. Copy this <targetSecret> file to appliance A.
9. **On appliance A**, enable appliance A as the source appliance. At the NetScaler prompt, type:

   enable ssl fipsSIMSource <targetSecret> <sourceSecret>
10. Copy this <sourceSecret> file to appliance B.
11. **On appliance B**, enable appliance B as the target appliance. At the NetScaler prompt, type:

    enable ssl fipsSIMtarget <keyVector> <sourceSecret>
12. **On appliance A**, create a FIPS key, as described in Creating a FIPS Key.
13. Export the FIPS key to the appliance's hard disk, as described in Exporting a FIPS Key.
14. Copy the FIPS key to the hard disk of the secondary appliance by using a secure file transfer utility, such as SCP.
15. **On appliance B**, import the FIPS key from the hard disk into the HSM of the appliance, as described in Importing an Existing FIPS Key.

To configure FIPS appliances in a high availability setup by using the configuration utility

1. On the appliance to be configured as the source appliance, in the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Info** tab, click **Enable SIM**.
3. In the **Enable HA Pair for SIM** dialog box, in the **Certificate File Name** text box, type the file name, with the path to the location at which the FIPS certificate should be stored on the source appliance.
4. In the **Key Vector File Name** text box, type the file name, with the path to the location at which the FIPS key vector should be stored on the source appliance.
5. In the **Target Secret File Name** text box, type the location for storing the secret data on the target appliance.
6. In the **Source Secret File Name** text box, type the location for storing the secret data on the source appliance.
7. Click **OK**. The FIPS appliances are now configured in HA mode.
8. Create a FIPS key, as described in Creating a FIPS Key. The FIPS key is automatically transferred from the primary to the secondary.

Example

In the following example, source.cert is the certificate on the source appliance, stored in the default directory, /nsconfig/ssl. This certificate must be transferred to the same location (/nsconfig/ssl) on the target appliance. The file target.secret is created on the target appliance and copied to the source appliance. The file source.secret is created on the source appliance and copied to the target appliance.

On the source appliance

```
init fipsSIMsource /nsconfig/ssl/source.cert
```

On the target appliance

```
init fipsSIMtarget/nsconfig/ssl/source.cert /nsconfig/ssl/target.key /nsconfig/ssl/target.secret
```

On the source appliance

```
enable fipsSIMsource /nsconfig/ssl/target.secret /nsconfig/ssl/source.secret
```

On the target appliance

```
enable fipsSIMtarget /nsconfig/ssl/target.key /nsconfig/ssl/source.secret
```

On the source appliance

```
create fipskey fips1 -modulus 2048 -exponent f4
export fipskey fips1 -key /nsconfig/ssl/fips1.key
```

Copy this key into the hard disk of the target appliance.

On the target appliance

```
import fipskey fips1 -key /nsconfig/ssl/fips1.key
```

The following diagram summarizes the transfer process

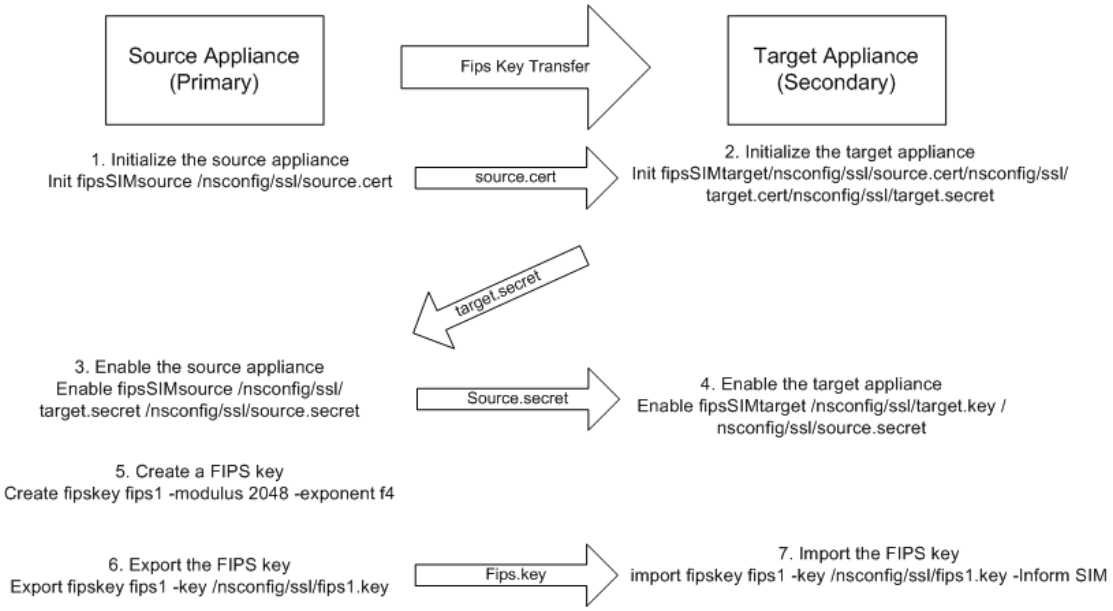


Figure 1. Transferring the FIPS Key-Summary

Updating the Firmware Version on a FIPS Card

You can use an update command to update the firmware version on the NetScaler 9950 FIPS or 9010 FIPS card from 4.6.0 to 4.6.1. The update command checks for compatible versions of firmware before starting the update. The command can be executed only by using the NetScaler command line.

For successful SIM key propagation from primary to secondary in an HA pair, the Cavium firmware version on each appliance should be identical. Perform the firmware update on the secondary appliance first. If executed on the primary appliance first, the long-running update process causes a failover.

Precautions

Before executing the firmware update command, review the following list of cautions:

- This process is only for updating from Cavium firmware version 4.6.0 to version 4.6.1. It is NOT possible to update from an earlier firmware version, such as version 4.3.5.
- The firmware update is possible on NetScaler 9.1 build 9.1_98.5 and above, and on all NetScaler 9.2 and 9.3 builds.
- The update fips -fipsFW 4.6.1 command can be successfully executed only once. After the firmware has been upgraded, subsequent attempts to issue the command do not have any effect.
- When the update is complete, the firmware cannot be downgraded to an earlier version.
- The update must be performed when the appliance is offline and network traffic is not passing through the appliance.
- Citrix recommends executing the update command through the serial console. If the command is executed from a Telnet or PuTTY session, the session might time out before command execution is complete.

To update the FIPS firmware version on a standalone NetScaler

1. Log on to the NetScaler, using the administrator credentials.
2. At the prompt, type show fips to confirm that the firmware version is 4.6.0.

   show fips

   ```
   FIPS HSM Info:
         HSM Label           : citrix-1
         Initialization      : FIPS-140-2 Level-2
         HSM Serial Number   : 8005535
         Firmware Version    : 4.6.0
         Total Flash Memory  : 14286412
         Free Flash Memory   : 14285704
         Total SRAM Memory   : 17038624
         Free SRAM Memory    : 17037184
   ```

3. Disable any monitors and save the configuration. At the prompt, type:

   - disable lb mon <servicename>
   - save config

4. Perform the update. At the prompt, type:

   update fips -fipsFW 4.6.1

   You are prompted to confirm that the command should be executed:

   ```
   This command will update compatible version of the FIPS
   firmware from 4.6.0 to 4.6.1. You must save the current
   configuration (save config) before executing this command. You
   must reboot the system after execution of this command, for the
   firmware update to take effect.
   Do you want to continue?(Y/N)y
   Done.
   ```

   During execution, the command performs the following clean-up:

   - Flushes all SSL connections.
   - Stops SSL card monitoring. Current card status is maintained.
   - Prevents all SSL virtual servers, services, and secure monitoring from accepting or initiating new connections. The following error counter appears:

ssl_err_down_for_cfg_change.

The update takes from three to five seconds. The update command is blocking, which means that no other actions are executed until the command finishes. The command prompt is displayed only after execution of the command is completed.

5. Restart the appliance. At the prompt, type:

   reboot

6. Verify that the update is successful. At the prompt, type:

   show fips

   The firmware version displayed in the output should be 4.6.1.

   If the appliance is not restarted or the update is unsuccessful, any subsequent commands related to FIPS are stopped and the following error message appears:

   ERROR: Operation not permitted - FIPS card firmware update
   done, please reboot the system

   Restart the appliance and reissue the update fips -fipsFW 4.6.1 command.

7. Enable the monitors that were disabled in step 3. At the prompt, type:

   enable lb mon <servicename>

To update the FIPS firmware version on NetScaler appliances in a high availability pair

1. Log on to the secondary node and perform the update as described in To update the FIPS firmware version on a standalone NetScaler.

   Force the secondary node to become primary. At the prompt, type:

   force failover

   A confirmation prompt appears:

   Please confirm whether you want force-failover (Y/N)? [N]:y
   Done

2. Log on to the new secondary node (old primary) and perform the update as described in To update the FIPS firmware version on a standalone NetScaler.

3. Force the new secondary node to become primary again. At the prompt, type:

   force failover

   A confirmation prompt appears:

   Please confirm whether you want force-failover (Y/N)? [N]:y
   Done

Resetting a Locked HSM

The HSM becomes locked (no longer operational) if you change the SO password, restart the appliance without saving the configuration, and make three unsuccessful attempts to change the password. This is a security measure for preventing unauthorized access attempts and changes to the HSM settings.

Important: To avoid this situation, save the configuration after initializing the HSM.

If the HSM is locked, you must reset the HSM to restore the default passwords. You can then use the default passwords to access the HSM and configure it with new passwords. When finished, you must save the configuration and restart the appliance.

Caution: Do not reset the HSM unless it has become locked.

To reset a locked HSM by using the NetScaler command line

At the NetScaler command prompt, type the following commands to reset and re-initialize a locked HSM:

- reset fips
- set ssl fips -**initHSM Level-2** <new SO password> <old SO password> <user password> [-**hsmLabel** <string>]
- saveconfig
- reboot [-warm]

Example

```
reset fips
set fips -initHSM Level-2 newsopin123 sopin123 userpin123 -hsmLabel NSFIPS
saveconfig
reboot -warm
```

Note: The SO and User passwords are the default passwords.

To reset a locked HSM by using the configuration utility

1. In the navigation pane, expand **SSL**, and then click **FIPS**.
2. In the details pane, on the **FIPS Info** tab, click **Reset FIPS**.
3. Configure the HSM, as described in Configuring the HSM.
4. In the details pane, click **Save**.

FIPS Approved Algorithms and Ciphers

The FIPS approved algorithms are:

Key-Exchange algorithms

- RSA

Cipher algorithms

- SSL3-DES-CBC-SHA (9010 FIPS and 9950 FIPS appliances only)
- SSL3-DES-CBC3-SHA
- TLS1-AES-256-CBC-SHA
- TLS1-AES-128-CBC-SHA

Note: RC4 (ARC4) is not a FIPS-approved algorithm.

SSL virtual server is marked UP only when default ciphers (FIPS) are configured. On the 9010 FIPS and 9950 FIPS appliances, to enable other ciphers on an SSL virtual server, use the following command:

set ssl Vserver [-**nonfipscipher** (ENABLE|DISABLE)]

SSL3-RC4-SHA is the only non-FIPS-approved cipher supported on the 9010 and 9950 FIPS appliances. You can create cipher groups of FIPS-approved ciphers and SSL3-RC4-SHA ciphers only on the 9010 and 9950 FIPS appliances.

String Maps

You can use string maps to perform pattern matching in all NetScaler features that use the default policy syntax. A string map is a NetScaler entity that consists of key-value pairs. The keys and values are strings in either ASCII or UTF-8 format. String comparison uses two new functions, `MAP_STRING(<string_map_name>)` and `IS_STRINGMAP_KEY(<string_map_name>)`.

A policy configuration that uses string maps performs better than one that does string matching through policy expressions, and you need fewer policies to perform string matching with a large number of key-value pairs. String maps are also intuitive, simple to configure, and result in a smaller configuration.

How String Maps Work

String maps are similar in structure to pattern sets (a pattern set defines a mapping of index values to strings; a string map defines a mapping of strings to strings) and the configuration commands for string maps (commands such as add, bind, unbind, remove, and show) are syntactically similar to configuration commands for pattern sets. Also, as with index values in a pattern set, each key in a string map must be unique across the map. The following table illustrates a string map called url_string_map, which contains URLs as keys and values.

Table 1. String Map "url_string_map"

Key	Value
/url_1.html	http://www.redirect_url_1.com/url_1.html
/url_2.html	http://www.redirect_url_2.com/url_2.html
/url_3.html	http://www.redirect_url_1.com/url_1.html

The following table describes the two functions that have been introduced to enable string matching with keys in a string map. String matching is always performed with the keys. Additionally, the following functions perform a comparison between the keys in the string map and the complete string that is returned by the expression prefix. The examples in the descriptions refer to the preceding example.

Table 2. String Map Functions

nction	Description

EXT>.MAP_STRING(<string_map_name>)	Checks whether the value returned by the expression prefix TEXT matches a in the string map, and returns the value that corresponds to the key. If no ke map matches the value returned by the expression prefix, the function return string. The IGNORECASE and NOIGNORECASE functions can be used for case and case-sensitive comparison, respectively. **Example 1:** HTTP.REQ.URL.MAP_STRING("url_string_map") checks wh string returned by HTTP.REQ.URL is a key in the string map url_string_map. of HTTP.REQ.URL is /url_1.html, the function returns http://www.redirect_url_1.com/url_1.html. **Example 2:** HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).MAP_STRING("url_str checks whether the string returned by HTTP.REQ.URL is a key in the string r url_string_map. The comparison does not consider case. If the string returned HTTP.REQ.URL is /URL_1.html, the function returns http://www.redirect_url_1.com/url_1.html. **Parameters:** string_map_name - The string map.
EXT>.IS_STRINGMAP_KEY(<string_map_name>)	Returns TRUE if the string returned by the expression prefix TEXT is a key in map. The IGNORECASE and NOIGNORECASE functions can be used for case- and case-sensitive string matching, respectively. **Example 1:** HTTP.REQ.URL.IS_STRINGMAP_KEY("url_string_map") returns TRUE HTTP.REQ.URL is one of the keys in url_string_map. **Example 2:** HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE). IS_STRINGMAP_KEY("url_string_map") returns TRUE if the value of HT is one of the keys in url_string_map. In this case, key lookup does not conside Therefore, the function returns TRUE even if the value of HTTP.REQ.URL is /URL_3.html. **Parameters:** string_map_name - The string map.

How String Maps Work

String maps are similar in structure to pattern sets (a pattern set defines a mapping of index values to strings; a string map defines a mapping of strings to strings) and the configuration commands for string maps (commands such as add, bind, unbind, remove, and show) are syntactically similar to configuration commands for pattern sets. Also, as with index values in a pattern set, each key in a string map must be unique across the map. The following table illustrates a string map called url_string_map, which contains URLs as keys and values.

Table 1. String Map "url_string_map"

Key	Value
/url_1.html	http://www.redirect_url_1.com/url_1.html
/url_2.html	http://www.redirect_url_2.com/url_2.html
/url_3.html	http://www.redirect_url_1.com/url_1.html

The following table describes the two functions that have been introduced to enable string matching with keys in a string map. String matching is always performed with the keys. Additionally, the following functions perform a comparison between the keys in the string map and the complete string that is returned by the expression prefix. The examples in the descriptions refer to the preceding example.

Table 2. String Map Functions

nction	Description

EXT>.MAP_STRING(<string_map_name>)	Checks whether the value returned by the expression prefix TEXT matches a in the string map, and returns the value that corresponds to the key. If no ke map matches the value returned by the expression prefix, the function return string. The IGNORECASE and NOIGNORECASE functions can be used for case and case-sensitive comparison, respectively. **Example 1:** HTTP.REQ.URL.MAP_STRING("url_string_map") checks wh string returned by HTTP.REQ.URL is a key in the string map url_string_map. of HTTP.REQ.URL is /url_1.html, the function returns http://www.redirect_url_1.com/url_1.html. **Example 2:** HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).MAP_STRING("url_str checks whether the string returned by HTTP.REQ.URL is a key in the string r url_string_map. The comparison does not consider case. If the string returned HTTP.REQ.URL is /URL_1.html, the function returns http://www.redirect_url_1.com/url_1.html. **Parameters:** string_map_name - The string map.
EXT>.IS_STRINGMAP_KEY(<string_map_name>)	Returns TRUE if the string returned by the expression prefix TEXT is a key in map. The IGNORECASE and NOIGNORECASE functions can be used for case- and case-sensitive string matching, respectively. **Example 1:** HTTP.REQ.URL.IS_STRINGMAP_KEY("url_string_map") returns TRUE HTTP.REQ.URL is one of the keys in url_string_map. **Example 2:** HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE). IS_STRINGMAP_KEY("url_string_map") returns TRUE if the value of HT is one of the keys in url_string_map. In this case, key lookup does not conside Therefore, the function returns TRUE even if the value of HTTP.REQ.URL is /URL_3.html. **Parameters:** string_map_name - The string map.

Configuring a String Map

You first create a string map and then bind key-value pairs to it. You can create a string map from the command-line interface (CLI) or the configuration utility. In the CLI, you first use the `add policy stringmap` command to create a string map. You then use the `bind policy stringmap` command to bind key-value pairs, one pair at a time. In the configuration utility, you create a string map and bind key-value pairs to it from a single dialog box.

To create a string map by using the NetScaler command line

At the NetScaler command prompt, type:

`add policy stringmap` `<name> [`**`-comment`** `<string>]`

Example

```
> add policy stringmap url_string_map
Done
```

To bind a key-value pair to the string map by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a key-value pair to a string map and verify the configuration:

- **`bind policy stringmap`** `<name> <key> <value>`
- **`show policy stringmap`** `<name>`

Example

```
> bind policy stringmap url_string_map1 "/url_1.html" "http://www.redirect_url_1.com/url_1.html"
Done
> show policy stringmap url_string_map1
      String map:     url_string_map1

1)     Key:    /url_1.html     Value:  http://www.redirect_url_1.com/url_1.html
Done
>
```

After you create a string map, you can modify only the `comment` parameter from the command line. To modify a string map by using the NetScaler command line, type the `set policy stringmap` command followed by the name of the string map and the `comment` parameter with its new value.

Parameters for configuring a string map

name

Name of the string map. Maximum Length: 127.

key

The key in the string map. Maximum Length: 2047.

value

The value associated with the key in the string map. Maximum Length: 2047.

To create a string map by using the NetScaler configuration utility

1. In the navigation pane, expand **AppExpert**, and then click **String Maps**.
2. In the details pane, do one of the following:
   - To create a string map, click **Add**.
   - To modify a string map, click the name of the string map, and then click **Open**.
3. In the **Create String Map** dialog box, in the **Name** box, specify a name for the string map, and then do one of the following:
   - To add a key-value pair, click **Add**.
   - To modify a key-value pair, click the key-value pair, and then click **Open**.
4. In the **Create Mapping** or **Configure Mapping** dialog boxes, specify values for the following parameters, which correspond to parameters described in "Parameters for configuring a string map" as shown:
   - **Key***—key (cannot be changed for an existing key-value pair)
   - **Value***—value

   * A required parameter
5. Click **Create** or **OK**.
6. Click **Create** or **OK**.

String Maps Use Cases

You can use string maps in all features that support the newer default policy syntax. For example, string maps can be used in responder redirects and rewrite actions. You can also reuse a given string map in multiple features.

Use Case: Responder Policy With a Redirect Action

The following use case involves a responder policy with a redirect action. In the example below, the first four commands create the string map `url_string_map` and bind the three key-value pairs used in the earlier example. After creating the map and binding the key-value pairs, you create a responder action (`act_url_redirects`) that redirects the client to the corresponding URL in the string map or to `www.default.com`. You also configure a responder policy (`pol_url_redirects`) that checks whether requested URLs match any of the keys in `url_string_map` and then performs the configured action. Finally, you bind the responder policy to the content switching virtual server that receives the client requests that are to be evaluated.

```
add stringmap url_string_map

bind stringmap url_string_map /url_1.html
http://www.redirect_url_1.com/url_1.html

bind stringmap url_string_map /url_2.html
http://www.redirect_url_2.com/url_2.html

bind stringmap url_string_map /url_3.html
http://www.redirect_url_1.com/url_1.html

add responder action act_url_redirects redirect
'HTTP.REQ.URL.MAP_STRING("url_string_map") ALT "www.default.com"'
-bypassSafetyCheck yes

add responder policy pol_url_redirects TRUE act_url_redirects

bind cs vserver csw_redirect -policyname pol_url_redirects -priority
1 -type request
```

SureConnect

You can use the SureConnect feature of the Citrix® NetScaler® appliance to service all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.

When servers are overloaded with the requests, the servers might either respond slowly or not at all. The SureConnect feature enables the NetScaler appliance to detect and compensate such conditions by ensuring that every client request gets serviced in some way, such as either a custom Web page or actual content is sent to the client.

SureConnect is activated when the response time or maximum server connections to a client request exceeds a limit that you have set. The SureConnect browser window displays one of the following:

- A progress bar with the amount of time remaining until the requested content will be available.
- Alternate Web content of your choice (alternate page).
- Both a progress bar and alternate page.
- Complete custom content of your choice.

You can configure whether the SureConnect progress bar alone is displayed or both the progress bar and the alternate page are displayed.

When the server becomes responsive again, the original request for content is served. If the user chooses, the alternate content window can remain in focus.

Subsequent requests from the same user within the same session are served immediately. This can be configured using the settings described later in this section.

SureConnect can be activated when a response is delayed, and when the number of user connections to a given URL exceeds a specified threshold.

SureConnect works with all standard browsers, including Microsoft Internet Explorer, Netscape Navigator, and Mozilla Firefox.

SureConnect is advantageous in the following situations:

- **Full server queue**

  The server can respond fast, but there are too many users. This results in the server's queue being full and unable to process additional client requests.

  SureConnect Solution: In this situation, the SureConnect window is displayed, showing the time left until the content will be available. The alternate page is displayed under the progress bar, if an alternate page has been configured.

-

Large response delay

The server response is slow. Typically, if a Web server does not respond to a client request quickly, the user will leave the site.

SureConnect Solution: When the predicted delay reaches a configured time threshold, the SureConnect window displays the progress bar and the optional alternate page in the client browser.

- **Client time-out**

  When the client requests content from a very slow Web site, a time-out message displays in the client browser, and the content is not delivered. The user may leave the site.

  SureConnect Solution: The appliance stores the request until the server is no longer busy and delivers the requested content to the client.

- **Server experiencing a traffic surge**

  The server typically responds quickly, but the current load of open connections is greater than the server capacity to serve them. Therefore, the server response is delayed.

  SureConnect Solution: A SureConnect window is displayed in the client browser, showing the time left. The alternate page from the server is also displayed if it has been configured.

Installing SureConnect

SureConnect files must be installed on the alternate content server, which can be the same as the primary server.

On a Windows server, extract the sc_xx.exe file (where xx is the build number), or on a UNIX server, extract the sc_xx.tar file (where xx is the build number).

Note: You must install SureConnect in the default Web root directory.

If the alternate content server is the same as the primary server, place the SureConnect and alternate content files in any directory under the Web root directory. Specify this path when you add a policy to configure SureConnect. By default, SureConnect files are installed in the /Citrix NetScaler appliance directory under the default Web root directory.

If the alternate content server is different than the primary server, the SureConnect and alternate content files must be in a unique directory under the Web root directory. By default, this unique directory is the /Citrix NetScaler system directory. Specify this path when you add a policy to configure SureConnect.

The following files are extracted:

- Alternate content files (progressbar.htm, alternatepage.htm, and barandpage.htm)
- System-Logo.gif
- Customer-Logo.gif
- Sample.gif
- README.txt.

Installing SureConnect

SureConnect files must be installed on the alternate content server, which can be the same as the primary server.

On a Windows server, extract the sc_xx.exe file (where xx is the build number), or on a UNIX server, extract the sc_xx.tar file (where xx is the build number).

Note: You must install SureConnect in the default Web root directory.

If the alternate content server is the same as the primary server, place the SureConnect and alternate content files in any directory under the Web root directory. Specify this path when you add a policy to configure SureConnect. By default, SureConnect files are installed in the /Citrix NetScaler appliance directory under the default Web root directory.

If the alternate content server is different than the primary server, the SureConnect and alternate content files must be in a unique directory under the Web root directory. By default, this unique directory is the /Citrix NetScaler system directory. Specify this path when you add a policy to configure SureConnect.

The following files are extracted:

- Alternate content files (progressbar.htm, alternatepage.htm, and barandpage.htm)
- System-Logo.gif
- Customer-Logo.gif
- Sample.gif
- README.txt.

Installing on UNIX

This section describes how to install SureConnect alternate content on a UNIX server. The following are the prerequisites:

- The UNIX server is running the Apache server.
- The shell with the # prompt is in use.
- Apache is installed in the default location.
- The sc_xx.tar file is downloaded from the organization's Web site into the /var/ftp/incoming directory.

To install SureConnect

1. At the command prompt, navigate to the htdocs directory:

   cd /usr/local/apache/htdocs

2. Type the following command:

   tar xvpf/var/ftp/incoming/sc_xx.tar

The output from the .tar file is displayed. A /Citrix NetScaler system directory is created under the specified path and the SureConnect files are installed.

Installing on Windows

This section describes how to install SureConnect alternate content on a Windows server. The following are the prerequisites:

- The server is running the Microsoft® Internet Information Server.
- The DOS prompt is being used.
- The SureConnect zip (self-extracting) file is downloaded from the organization Web site using FTP into the C:\inetpub\wwwroot directory.

To install SureConnect on Windows

Do one of the following:

- At the command prompt, navigate to the wwwroot directory:

  cd c:\inetpub\wwwroot

- Type the name of the executable file:

  sc_xx.exe

- Double-click the sc_xx.exe icon from the Microsoft Windows Explorer® Web browser, extract from the compressed file into the default path (for example, the c:\inetpub\wwwroot directory).

Output from the zip file is displayed. A /Citrix NetScaler system directory is created under the specified path, and the SureConnect files are installed.

Configuring SureConnect

The following topics describe how to configure SureConnect for scenarios involving alternate server failure.

- Configuring the Response for Alternate Server Failure
- Configuring the SureConnect Policies
- Customizing the Alternate Content File
- Configuring SureConnect for Citrix NetScaler Features

Configuring the Response for Alternate Server Failure

If the alternate server fails, and the primary server cannot immediately deliver the requested content to the client, SureConnect does not display alternate content from the failed alternate server in the client Web browser.

The Citrix® NetScaler® appliance automatically sends a response to the client browser. You can customize the server response to display information suited to your needs.

The default response is:

```
Your Request is being processed… Estimated Time: ____ Secs
```

Customizing the Default Response

The NetScaler appliance automatically sends the response to the client if the alternate server fails, or if the appliance is configured to send the default response.

To customize the default response of the appliance, create a vsr.htm file (a sample is provided in this section) as follows:

- The file can contain any valid HTML statements other than embedded objects.
- The file size cannot exceed 800 bytes.
- The file must reside on the NetScaler appliance. If you have a high availability (HA) setup, the file must reside on the primary and secondary nodes. Any changes made to the file on the primary node must also be applied to the file on the secondary node.
- Put vsr.htm file in the /etc directory.

To customize the default response

Change any of the contents between the </HEAD> and </HTML> tags in the vsr.htm file. Following is the sample content from vsr.htm file. The sections that you can edit are in bold text.

HTTP/1.1 200 OK
Server: NS_WS3.0
Content-Type: text/html
Cache-control: no-cache
Pragma: no-cache
Set-Cookie: NSC_BPIP=@@SID@@; path=/
<HTML> <HEAD> <META HTTP-EQUIV="Refresh" CONTENT="0">
</HEAD> **Your request is being processed...**
**
Estimated Delay: @@DELAY@@ Sec ** </HTML>

Note: Include @@DELAY@@ to display the predicted delayed response time in seconds.

SureConnect with In-Memory response (NS action)

The following example explains how to configure the SureConnect feature so that the appliance supplies alternate content.

In this example, there are two physical servers with IP addresses 10.101.3.187 and 10.101.3.188. The appliance load balances both servers.

The file that contains the alternate content is vsr.htm. It is copied from the file system into system memory. Services are loaded until the SureConnect policy triggers, and the appliance supplies the alternate content.

To add a service by using the NetScaler command line

At the NetScaler command prompt, type:

```
enable feature SC LB
add service psvc1 10.101.3.187 http 80
add service psvc2 10.101.3.188 http 80
add vserver vs-NSact HTTP 10.101.3.201 80
bind lb vserver vs-NSact psvc1
bind lb vserver vs-NSact psvc2
add sc policy policyNS -url /cgi-bin/*.cgi -delay 400000
-action NS
set sc parameter -vsr /nsconfig/ssl/vsr.htm
bind lb vserver vs-NSact -policyName policyNS
set lb vserver vs-NSact -sc ON
save config
```

To add a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, in **Service Name** text box, type a name for the service (for example, **svc1**).
4. In **Server** and **Protocol**, select a server and a protocol (for example, **10.101.3.187** and **HTTP**).
5. In the **Port** text box, type the port number (for example, **80**).

Configuring the SureConnect Policies

You can configure the following SureConnect policies. The NetScaler appliance matches incoming requests in the order the policies are configured:

- Exact URL-based policies
- Wildcard rule-based policies

Configuring Exact URL-Based Policies

The NetScaler appliance matches an incoming request against the exact URL-based policy that has been configured.

For information about configuring SureConnect from the command-line interface, see the *Citrix NetScaler Command Reference Guide* at .

To configure an exact URL-based policy by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then click **SureConnect**.
2. In the details pane, click **Add**.
3. In the **Create SureConnect Policy** dialog box, in the **Name** text box, type the name of the policy.
4. By default, the **URL** option is selected. In **Value** text box, type the exact URL.
5. Under **Settings**, in **Delay (microseconds)**, type the number in micro seconds after which you want the policy to be evaluated (for example, **20**). The minimum value is 1 and maximum value is 599999999.
6. In **Maximum Client Connections**, type a number that denotes that the policy should be evaluated after the number of client connections reach the specified number (for example, **9999**). The minimum value is 1 and the maximum value is 0xFFFFFFFE.
7. Under **Action**, select one of the actions from the **Choose Action** drop-down list.
   - **ACS.** Specifies the alternate content to be served from the alternate server with the alternate content path.
   - **NS.** Specifies the alternate content to be served from the appliance.
   - **NOACTION.** Specifies that no alternate content is to be served.
8. The **Alternate Service Name** is active if you select **ACS** in the previous step. Select one of the service names from the drop-down list, and in **Alternate Content Path**, enter the path. The **Alternate Content Path** is the location where the alternate content is stored.
9. Click **Create**, and click **Close**. The URL-based policy appears in the right pane, and a message displays in the status bar that the policy is successfully configured.

Configuring Wildcard Rule-Based Policies

SureConnect matches the incoming requests to a defined rule, if you configure a rule-based policy.

To configure a SureConnect policy based on a wildcard rule by using NetScaler command line

1. Create the expression(s).

   Use the add expression command to create each expression.

2. Create the rule(s).

   Use the add sc policy command with the **-rule** expression_logic argument to specify the rule(s). In the **-rule** expression_logic argument, refer to the expression(s) you created in step 1.

   Repeat this command to create and name each rule.

The following example creates a rule “rule = = /*.cgi”:

```
add vserver vs-lb http 1.1.1.1 80
add expression expr1 url == /cgi-bin/*.cgi
add expression expr2 url == /index.html
add sc policy surecpolicy1 -rule (expr1||expr2) -delay 1000000 -action NS
bind lb vserver vs-lb -policyName surecpolicy1
```

To complete the SureConnect configuration, you will need to enter additional commands, beyond those shown in the example.

To configure a wildcard rule-based policy by using the configuration utility

1. In the navigation pane, expand **Protection Features**, and then click **SureConnect**.
2. In the details pane, click **Add**.
3. In the **Create SureConnect Policy** dialog box, in the **Name** text box, type the name of the policy.
4. Under **What to Monitor**, click **Expression**, and then click **Configure**.
5. In the **Create Expression** dialog box, click **Add**.
6. In the **Add Expression** dialog box, enter an expression. For example, you can select an **Expression Type** of General, a **Flow Type** of REQ, a **Protocol** of HTTP, a **Qualifier** of URLQUERY, an **Operator** of CONTAINS, and in the **Value** text box, type AA. For more information about expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .
7. Click **OK**, and click **Close**.
8. In the **Create Expression** dialog box, click **Create**.

Examples of wildcard rules:

"`/sports/*`" matches all URLs under `/sports`

"`/sports*`" matches all URLs whose prefix matches "`/sports`", starting at the beginning of the URL.

"`/*.jsp`" matches all URLs whose file extension is "`jsp`"

When configuring rule-based policies, first add the more specific rule-based policies, before adding more generic rules (for example, `add /cgi-bin/sports*.cgi before adding /cgi-bin/*.cgi`).

Displaying the Configured SureConnect Policy

To view the SureConnect policy that you have configured, at the NetScaler command prompt, enter the show sc policy command.

Customizing the Alternate Content File

When SureConnect activates, it can display alternate content from one of the following files that you have configured:

- **progressbar.htm.** Displays the progress information.
- **alternatepage.htm.** Displays an alternate page.
- **barandpage.htm.** Displays both the progress information and an alternate page.

The alternate content files are Javascript files. During SureConnect installation, these files are copied onto the server that contains the alternate content. These files can contain alternate content (including an alternate page) or references to other files that contain the alternate content.

This section describes the changes you can make to the alternate content file provided by the appliance.

```
//**** DEFINE YOUR VALUES HERE ****
var alt_url = "/Citrix NetScaler system /sample.gif";
var alt_url = "http://www.DomainName.com";
var Citrix NetScaler system _logo = "netscaler_logo.gif";
var our_logo = "netscaler_logo.gif";
var height = 450;
var width = 550;
var top = 200;
var left = 200;
var popunder = "no"; //specify yes for pop-under & no for pop-up
var shift_focus = "yes" //if you want to send pop-up to background on getting primary content else specify no
//**** YOUR DEFINITIONS ENDS HERE ****
```

You can make these changes:

- **var alt_url.** Specify the URL for the alternate content if a file provides the alternate content. For example:

  var alt_url = “/Citrix NetScaler system/sports.htm”

  Note: The alternate content file must be present in the /Citrix NetScaler system directory under the documents root of the Web server.
- **var our_logo.** Specify the image file of your organization logo.
- **var height.** Specify the height of the SureConnect window.
- **var width.** Specify the width of the SureConnect window.
- **var top and var left.** Specify the position of the SureConnect window.

- **var popunder.** Specifies the position of the alternate content window. Specify the value as NO to place the alternate content window above the original window. Specify the value as YES to place the alternate content window beneath the original window.

- **var shift_focus.** Specify the focus of the alternate content window. YES places the pop-up window in the background when getting the primary content. NO always keeps the pop-up window in focus, even when getting the primary content.

Note: For more information, see the README.txt file provided by the appliance with other alternate content files.

Configuring SureConnect for Citrix NetScaler Features

This section describes how SureConnect works in combination with the load balancing, content switching, cache redirection, and high availability features of the NetScaler appliance.

Configuring SureConnect for Load Balancing

You can use SureConnect in environments where the primary servers use the load balancing feature, with or without alternate servers. If the load balancing virtual server configured for SureConnect fails, the backup virtual server (if there is one) handles the traffic. Backup virtual servers do not support SureConnect policies.

Note: For information about load balancing, see the *Citrix NetScaler Traffic Management Guide* at .

Configuring SureConnect for Cache Redirection

You can use SureConnect in environments where cache redirection is configured. The primary server is a load balancing virtual server bound to the cache redirection virtual server. Regardless of any rules configured for the cache redirection feature:

- You can configure any URL for SureConnect.
- Once SureConnect is activated for a client, requests from the client are always sent to the origin server.

Configuring SureConnect for High Availability

SureConnect is compatible with NetScaler appliances operating in high availability mode.

Note: If the optional vsr.htm file is used, it must be present in both nodes (primary and secondary) and must use the same name and directory.

Activating SureConnect

You can set the Citrix® NetScaler® appliance to activate SureConnect if either of two criteria match. Both criteria are arguments to the add sc policy command, as described here:

-delay <microseconds>

The first time the client requests the URL, the appliance records how long the server takes to respond. The appliance will not activate SureConnect until the second time the URL is requested. The first and second requests may be from the same or different clients.

If you set -delay argument, SureConnect will be activated the second time the delay reaches the threshold you set.

-maxConn <positive_integer>

When the appliance receives a request, it checks the number of connections to the server for the configured URL. SureConnect is activated if the number of connections is greater than or equal to the value that you set for the **-maxConn** argument.

If you will be providing alternate content to be displayed in the client's Web browser, you should configure the **-action** argument of the add sc policy command. This specifies for the NetScaler appliance whether the alternate content is coming from a dedicated alternate server (**-action ACS**) or the appliance (**-action NS**).

When SureConnect is activated by the **-maxConn** argument, the SureConnect window and progress bar are displayed in the client's browser (with an alternate page, if configured).

SureConnect Environments

The following topics describe SureConnect environments.

- Primary and Alternate Servers
- Configuration Checklist
- Example Configurations

Primary and Alternate Servers

The SureConnect environment uses a dedicated server to provide alternate content when the requested content is not available. The alternate content may include an alternate page, plus optional components such as frame set, organization logo, and so on. The alternate and primary servers can be the same server.

You can configure SureConnect to display a progress bar when the requested content is not available (or the progress bar and an alternate page).

The following figure illustrates the SureConnect environment.

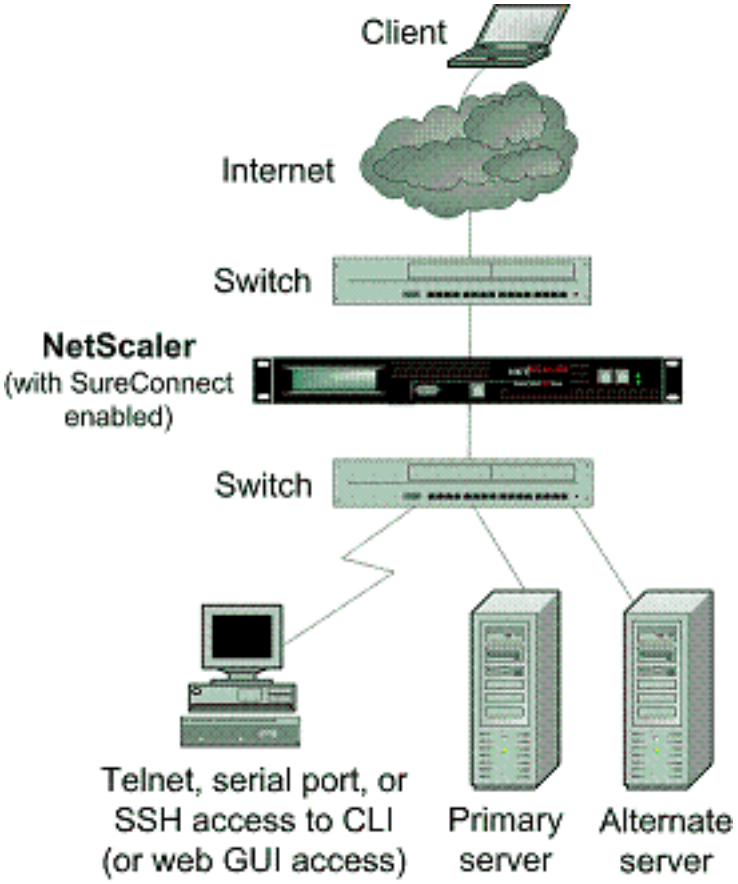


Figure 1. SureConnect - Primary and Alternate Servers

Configuration Checklist

Complete the following checklist before you start configuration:

Table 1. Configuration Checklist

☐	The same builds are running for the appliance and for the SureConnect files as suggested by appliance staff. Appliance Build Number: _____ SureConnect (sc_xx.exe) Build Number: _____
☐	The latest SureConnect files (style files) are extracted to: • All primary servers (required for NS action). • The alternate content server (required for ACS action).
☐	All customizations to the latest style and vsr.htm files are applied.
☐	The alternate content server is accessible from the Internet (required for ACS action).
☐	If the -redirectURL URL argument of the add vserver CLI command needs to be specified: • The URL is up and running. • This URL is not on the configured servers. • This URL does not match any content in the vserver (that is, do not redirect a missing URL to itself). Redirecting a missing URL to itself can send some browsers into an infinite loop.
☐	All URLs to be configured for SureConnect are top-level URLs only. (Only the URLs that occupy the whole window or frame can be configured, not the embedded objects).

Following are the steps to configure SureConnect in a setup with a primary server and a dedicated alternate server:

- Enable the SureConnect feature
- Add the SureConnect policy
- Bind the SureConnect policy

You can optionally configure the following:

- Redirect the client to another URL if the primary server fails, or send a customized response to the client if the alternate server fails.
- If the servers do not provide alternate content, send a default or customized response.

To redirect the client to another URL

1. Enable the SureConnect feature.

2. Define the primary server and its service.

   You must identify the original server for which SureConnect support is being configured. At the NetScaler command prompt, type the following command:

   add service <serviceName> <IP> HTTP <port>

   where <serviceName> assigns a name for the service; <IP> is the server's IP address; and <port> is the port number that the service will use.

   Repeat use of the add service CLI command for each service that is to be added.

   You can also configure SureConnect on a load balancing virtual server. At the NetScaler command prompt, type the following command:

   add vserver <name> HTTP <IP> <port>

3. Define and bind the SureConnect policy as follows. If you are configuring a rule-based policy, perform this step as described in Configuring Wildcard Rule-Based Policies. To configure a URL-based policy, at the NetScaler command prompt, type the following command:

   add sc policy <name> [**-url** <URL>] [**-delay** <microsec>] [**-maxConn** <positiveInteger>]

   For a detailed description of the add sc policy command, see the *Citrix NetScaler Command Reference Guide* at http://support.citrix.com/article/CTX123854.

   To bind the SureConnect policy, at the NetScaler command prompt, type the following command:

   bind service <serviceName> **-policyname** <string>

   where <serviceName> is the name of the service defined in step 2, and <string> is the name of the SureConnect policy.

   Repeat the bind service command for each policy created.

   You must include the alternate content page in the altContSvcName argument, and in the altContPath argument of the add sc policy command.

   In the following example, the name of the alternate content file is /Citrix NetScaler system /barandpage.htm, and this file resides in svc2.

4. To save the configuration, at the NetScaler command prompt, type the following command:

   save config

Example Configurations

The following examples illustrate various SureConnect configurations.

The examples assume that monitoring of physical services is enabled. If the alternate system is down, SureConnect will deliver the alternate content from the system itself.

Example 1 - SureConnect Progress Bar and Alternate Page

You can configure SureConnect to display both the progress bar and an alternate page to the user.

To bind a SureConnect policy to a load balancing virtual server, at the NetScaler command prompt, type the following commands:

bind lb vserver <virtualServerName> **-policyName** <string>

where <virutalServerName> is the name of the load balancing virtual server defined in step 2 of the configuration process, and <string> is the name of the SureConnect policy defined in step 3.

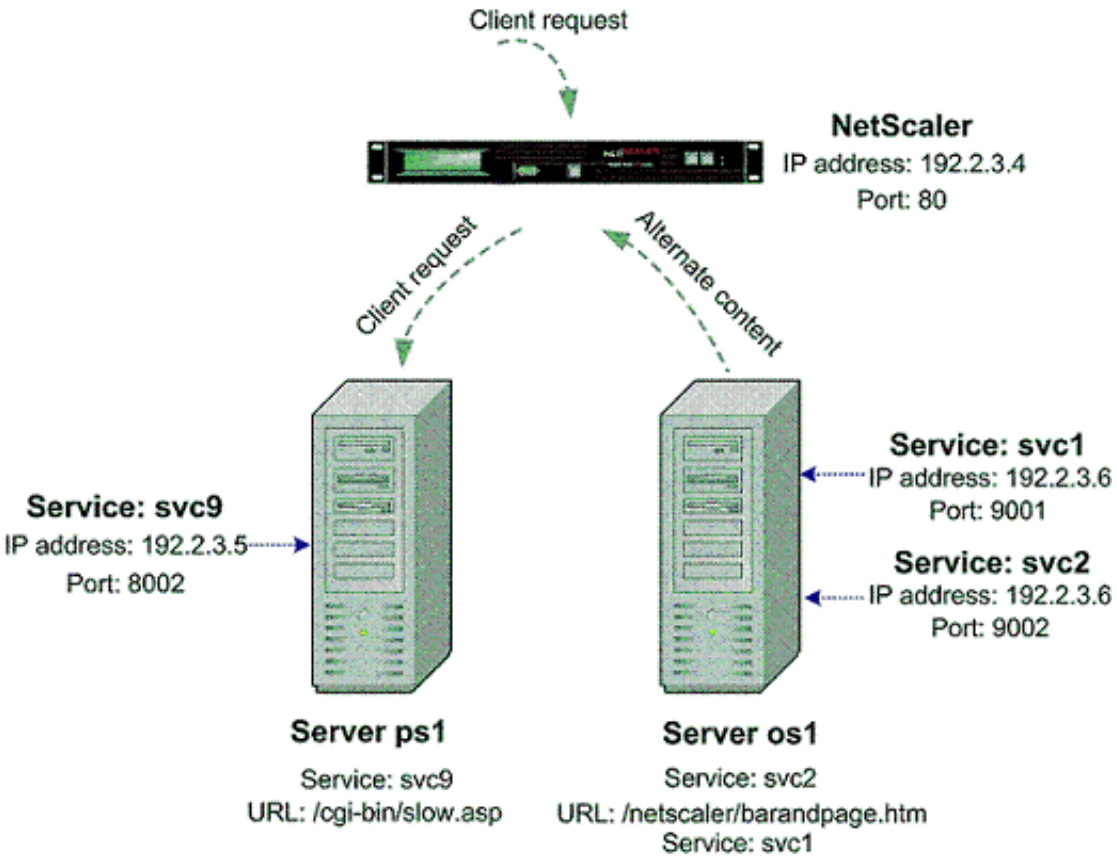


Figure 1. SureConnect Configuration - Example 1

At the NetScaler command prompt, type the following commands:

```
enable feature SC
show ns info
add service svc2 192.2.3.6 HTTP 9002
show server
show service svc2
add service svc9 192.2.3.5 HTTP 8002
add sc policy policy8 -url /cgi-bin/slow.asp
-delay 3000000 -action ACS svc2 /NetScaler 9000 system barandpage.htm
bind service svc9 -policyname policy8
set service svc9 -sc ON
save config
```

After you configure SureConnect, you can enter commands that show information to verify what you have configured.

Example 2 - SureConnect Progress Bar Only

In this example, SureConnect will display only the progress bar. The server orgsrvr with IP address 10.101.8.187 has service orgsvc. This server is connected to the appliance. The service is bound to the appliance. The progressbar.htm file specifies that only the progress bar will be displayed.

At the NetScaler command prompt, type the following commands:

```
enable feature SC
add service orgsvc 10.101.3.187 HTTP 80
add sc policy policy9 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS orgsvc /NetScaler 9000 system / progressbar.htm
bind service orgsvc -policyname policy9
set service orgsvc -sc ON
save config
```

Example 3 - SureConnect with Load Balancing

This example illustrates how to configure the load balancing feature so that SureConnect will display alternate contents from the primary server. For information about load balancing, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

In this example, two physical servers with IP 10.101.3.187 and 10.101.3.188 are being load balanced by the appliance. The name and location of the alternate page file is specified in the file alternatepage.htm, which resides on both servers.

The appliance has one configured virtual server address: 10.101.3.201. At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add vserver vs-SureC HTTP 10.101.3.201 80
bind lb vserver vs-SureC psvc1
bind lb vserver vs-SureC psvc2
add sc policy policy9 -url /cgi-bin/slow.asp -delay 4000000
-action ACS vs-SureC /NetScaler system /alternatepage.htm
bind lb vserver vs-SureC -policyName policy9
set lb vserver vs-SureC -sc ON
save config
```

Example 4 - SureConnect with Load Balancing (ACS Action)

This example illustrates how to configure the NetScaler appliance load balancing feature so that SureConnect will display alternate content from the alternate server. For information about load balancing, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

In this case, there are two physical servers, IP 10.101.3.187 and 10.101.3.188. Both are being load balanced by the appliance.

The name and location of the alternate page file are specified in file barandpage.htm, which resides on a third server not being load balanced.

The third server's IP address is 10.101.3.189. Because barandpage.htm is specified, the progress bar and alternate page will both be displayed.

The appliance has one configured virtual server "vsvr" whose IP address (Virtual Server) is 10.101.3.200.

At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add service alt-cont-svc 10.101.3.189 HTTP 80
add vserver vsvr HTTP 10.101.3.200 80
bind lb vserver vsvr psvc1
bind lb vserver vsvr psvc2
add sc policy policy10 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS alt-cont-svc
/NetScaler 9000 system /barandpage.htm
bind lb vserver vsvr -policyName policy10
set lb vserver vsvr -sc ON
save config
```

Example 5 - SureConnect with Content Switching

This example illustrates how to configure SureConnect where the NetScaler content switching and load balancing features are being used. SureConnect is configured on a load balancing virtual server bound to a content switching virtual server.

The alternate content is distributed under the content switching virtual server according to the content switching rules. For more information about load balancing and content switching, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX123869.

In this case, three physical services with IP addresses 10.100.100.104, 10.100.100.105, and 10.100.100.106 are bound to three load balancing virtual servers with IP addresses 10.100.100.101, 10.100.100.102, and 10.100.100.103. These three load balancing virtual servers are bound to a content switching virtual server with IP address 10.100.100.100.

In this setup, **lbvip1** contains .cgi content, **lbvip2** contains .gif content, and .**lbvip3** contains .html content.

The name and location of the alternate page file is specified in the file alternatepage.htm, which resides on **lbvip3**. The embedded objects in this file must be distributed according to the content switching rules (any embedded gif will reside on **lbvip2**, any embedded htm will reside on **lbvip3**, and so on).

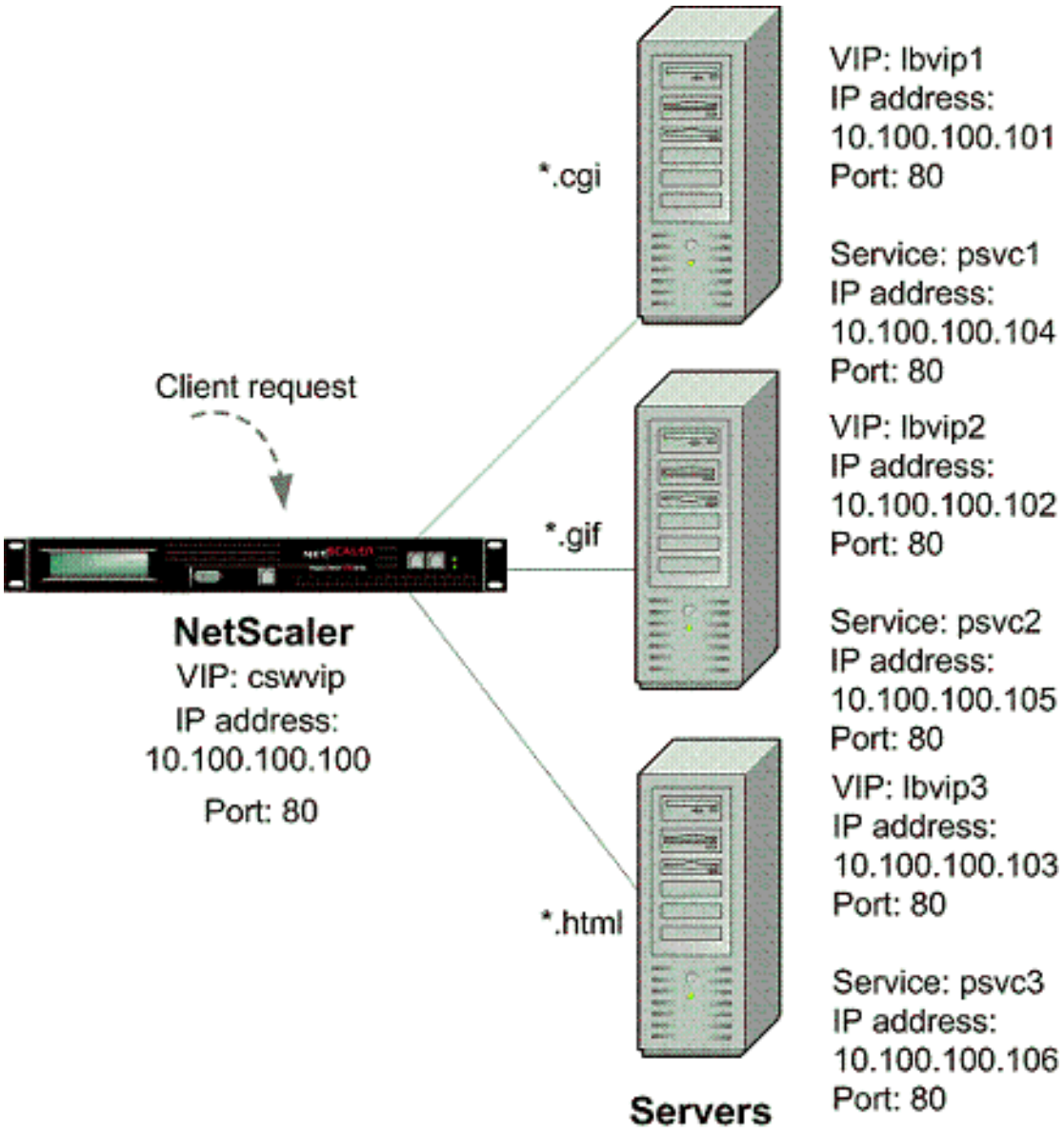


Figure 2. SureConnect Configuration - Example 5

At the NetScaler command prompt, type the following commands:

```
enable feature CS LB SC
add vserver cswvip HTTP 10.100.100.100 80 -type CONTENT
add vserver lbvip1 HTTP 10.100.100.101 80 -type ADDRESS
add vserver lbvip2 HTTP 10.100.100.102 80 -type ADDRESS
add vserver lbvip3 HTTP 10.100.100.103 80 -type ADDRESS
add service psvc1 10.100.100.104 HTTP 80
add service psvc2 10.100.100.105 HTTP 80
add service psvc3 10.100.100.106 HTTP 80
bind lb vserver lbvip1 psvc1
bind lb vserver lbvip2 psvc2
bind lb vserver lbvip3 psvc3
add cs policy CSWpolicy1 -url /*.cgi
```

```
bind cs vserver cswvip lbvip1 -policyName CSWpolicy1
add cs policy CSWpolicy2 -url /*.gif
bind cs vserver cswvip lbvip2 -policyName CSWpolicy2
add cs policy CSWpolicy3 -url /*.htm
bind cs vserver cswvip lbvip3 -policyName CSWpolicy3
add sc policy SCpol -url /cgi-bin/delay.cgi -delay 4000000 -action ACS cswvip /alternatepage.htm
bind lb vserver lbvip1 -policyName SCpol
set lb vserver lbvip1 -sc ON
save config
```

Surge Protection

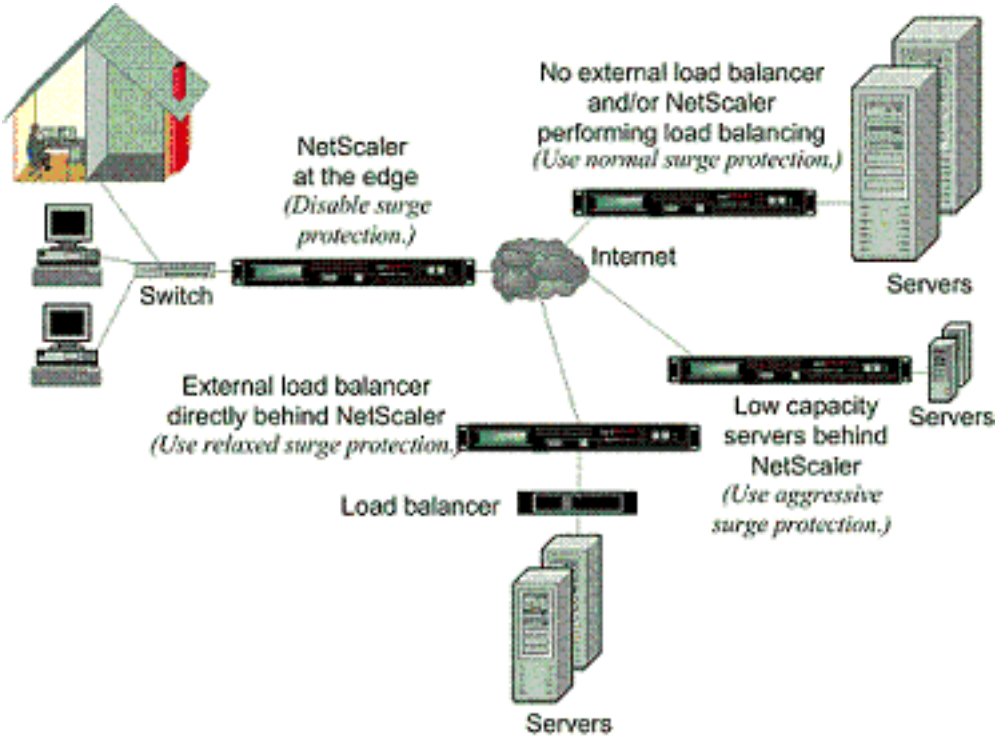
When a surge in client requests overloads a server, server response becomes slow, and the server is unable to respond to new requests. The Surge Protection feature ensures that connections to the server occur at a rate that the server can handle. The response rate depends on how surge protection is configured. The NetScaler appliance also tracks the number of connections to the server, and uses that information to adjust the rate at which it opens new server connections.

Surge protection is enabled by default. If you do not want to use surge protection, as will be the case with some special configurations, you must disable it.

The default surge protection settings are sufficient for most uses, but you can configure surge protection to tune it for your needs. First, you can set the throttle value to tell it how aggressively to manage connection attempts. Second you can set the base threshold value to control the maximum number of concurrent connections that the NetScaler appliance will allow before triggering surge protection. (The default base threshold value is set by the throttle value, but after setting the throttle value you can change it to any number you want.)

The following figure illustrates how surge protection is configured to handle traffic to a Web site.

Figure 1. A Functional Illustration of NetScaler Surge Protection



Note: If the NetScaler appliance is installed at the edge of the network, where it interacts with network devices on the client side of the Internet, the surge protection

feature must be disabled. Surge protection must also be disabled if you enable USIP (Using Source IP) mode on your appliance.

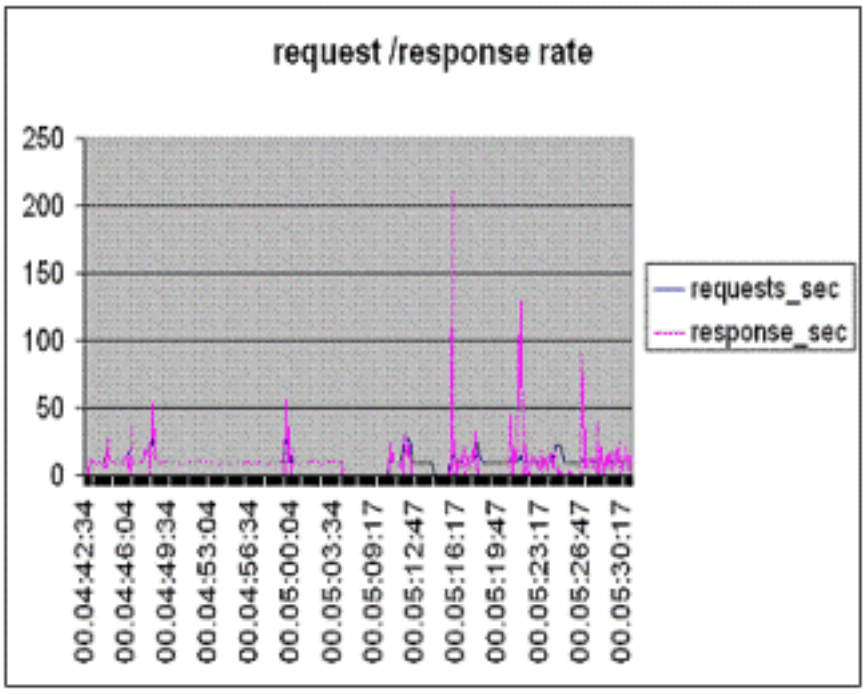
The following example and illustration show the request and response rates for two cases. In one case, surge protection is disabled, and in the other it is enabled.

When surge protection is disabled and a surge in requests occurs, the server accepts as many requests as it can process concurrently, and then begins to drop requests. As the server becomes more overloaded, it goes down and the response rate is reduced to zero. When the server recovers from the crash, usually several minutes later, it sends resets for all pending requests, which is abnormal behavior, and also responds to new requests with resets. The process repeats for each surge in requests. Therefore, a server that is under DDoS attack and receives multiple surges of requests can become unavailable to legitimate users.

When surge protection is enabled and a surge in requests occurs, surge protection manages the rate of requests to the server, sending requests to the server only as fast as the server can handle those requests. This enables the server to respond to each request correctly in the order it was received. When the surge is over, the backlogged requests are cleared as fast as the server can handle them, until the request rate matches the response rate.

The following figure compares the request and response scenarios when surge protection is enabled to that when it is disabled.

Figure 2. Request/Response Rate with and without Surge Protection



Disabling and Reenabling Surge Protection

The surge protection feature is enabled by default. When surge protection is enabled, it is active for any service that you add.

To disable or reenable surge protection by using the NetScaler command line

At the NetScaler command prompt, type one of the following sets of commands to disable or reenable surge protection and verify the configuration:

- disable ns feature SurgeProtection
- show ns feature
- enable ns feature SurgeProtection
- show ns feature

Example

```
disable ns feature SurgeProtection
 Done show ns feature

      Feature                  Acronym          Status
     disable ns feature -------                -------          ------
 1)    Web Logging              WL               ON
 2)    Surge Protection         SP               OFF
 .
 .
 .
 23)   HTML Injection           HTMLInjection    ON
 24)   NetScaler Push           push             OFF
 Done

 enable ns feature SurgeProtection
 Done
> show ns feature

      Feature                  Acronym          Status
      -------                 -------          ------
 1)    Web Logging              WL               ON
 2)    Surge Protection         SP               ON
```

```
.
.
.

23)  HTML Injection          HTMLInjection      ON
24)  NetScaler Push          push               OFF
Done
>
```

To disable or reenable surge protection by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change Advanced Features**.
3. In the **Configure Advanced Features** dialog box, clear the selection from the **Surge Protection** check box to disable the surge protection feature, or select the check box to enable the feature.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the feature has been enabled or disabled.

To disable or reenable surge protection for a particular service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then select **Services**. The list of configured services is displayed in the details pane.
2. In the details pane, select the service for which you want to disable or reenable the surge protection feature, and then click **Open**.
3. In the **Configure Service dialog box**, click the **Advanced** tab and scroll down.
4. In the **Others** frame, clear the selection from the **Surge Protection** check box to disable the surge protection feature, or select the check box to enable the feature.
5. Click **OK**. A message appears in the status bar, stating that the feature has been enabled or disabled.

   Note: Surge protection works only when both the feature and the service setting are enabled.

Disabling and Reenabling Surge Protection

The surge protection feature is enabled by default. When surge protection is enabled, it is active for any service that you add.

To disable or reenable surge protection by using the NetScaler command line

At the NetScaler command prompt, type one of the following sets of commands to disable or reenable surge protection and verify the configuration:

- disable ns feature SurgeProtection
- show ns feature
- enable ns feature SurgeProtection
- show ns feature

Example

```
disable ns feature SurgeProtection
 Done show ns feature

      Feature                  Acronym          Status
     disable ns feature -------                -------          ------
 1)    Web Logging               WL              ON
 2)    Surge Protection          SP              OFF
 .
 .
 .
 23)   HTML Injection            HTMLInjection      ON
 24)   NetScaler Push            push             OFF
 Done

 enable ns feature SurgeProtection
 Done
> show ns feature

      Feature                  Acronym          Status
      -------                 -------          ------
 1)    Web Logging               WL              ON
 2)    Surge Protection          SP              ON
```

```
.
.
.

23)  HTML Injection          HTMLInjection      ON
24)  NetScaler Push          push               OFF
Done
>
```

To disable or reenable surge protection by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change Advanced Features**.
3. In the **Configure Advanced Features** dialog box, clear the selection from the **Surge Protection** check box to disable the surge protection feature, or select the check box to enable the feature.
4. Click **OK**.
5. In the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the feature has been enabled or disabled.

To disable or reenable surge protection for a particular service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then select **Services**. The list of configured services is displayed in the details pane.
2. In the details pane, select the service for which you want to disable or reenable the surge protection feature, and then click **Open**.
3. In the **Configure Service dialog box**, click the **Advanced** tab and scroll down.
4. In the **Others** frame, clear the selection from the **Surge Protection** check box to disable the surge protection feature, or select the check box to enable the feature.
5. Click **OK**. A message appears in the status bar, stating that the feature has been enabled or disabled.

   Note: Surge protection works only when both the feature and the service setting are enabled.

Setting Thresholds for Surge Protection

To set the rate at which the NetScaler appliance opens connections to the server, you must configure the threshold and throttle values for surge protection.

The following figure shows the surge protection curves that result from setting the throttle rate to relaxed, normal, or aggressive. Depending on the configuration of the server capacity, you can set base threshold values to generate appropriate surge protection curves.

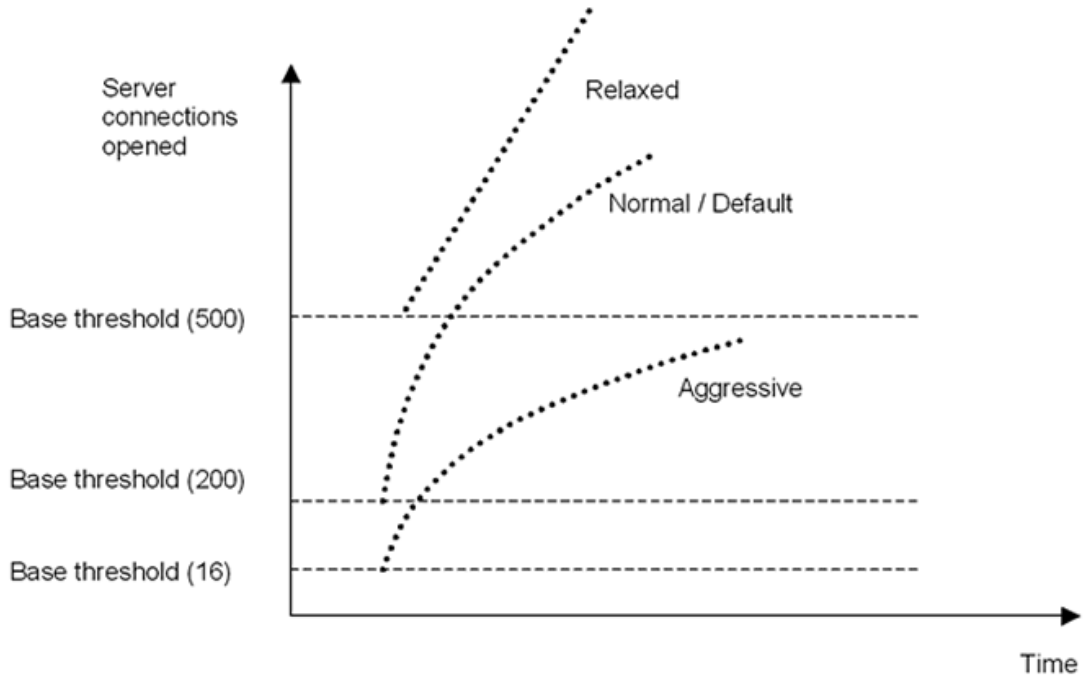


Figure 1. Surge Protection Curves

Your configuration settings affect the behavior of surge protection in the following manner:

- If you do not specify a throttle rate, it is set to normal (the default value), and the base threshold is set to 200, as shown in the preceding figure.
- If you specify a throttle rate (aggressive, normal, or relaxed) without specifying a base threshold, the curve reflects the default values of the base threshold for that throttle rate. For example, if you set the throttle rate to relaxed, the resulting curve will have the base threshold value of 500.
- If you specify only the base threshold, the entire surge protection curve shifts up or down, depending on the value you specify, as shown in the figure that follows.
- If you specify both a base threshold and a throttle rate, the resulting surge protection curve is based on the set throttle rate and adjusted according to the value set for the base threshold.

In the following figure, the lower curve (Aggressive 1) results when the throttle rate is set to aggressive but the base threshold is not set. The upper curve (Aggressive 2) results when the base threshold is set to 500, but the throttle rate is not set. The second upper curve (Aggressive 2) also results when the base threshold is set to 500, and the throttle rate is set to aggressive.

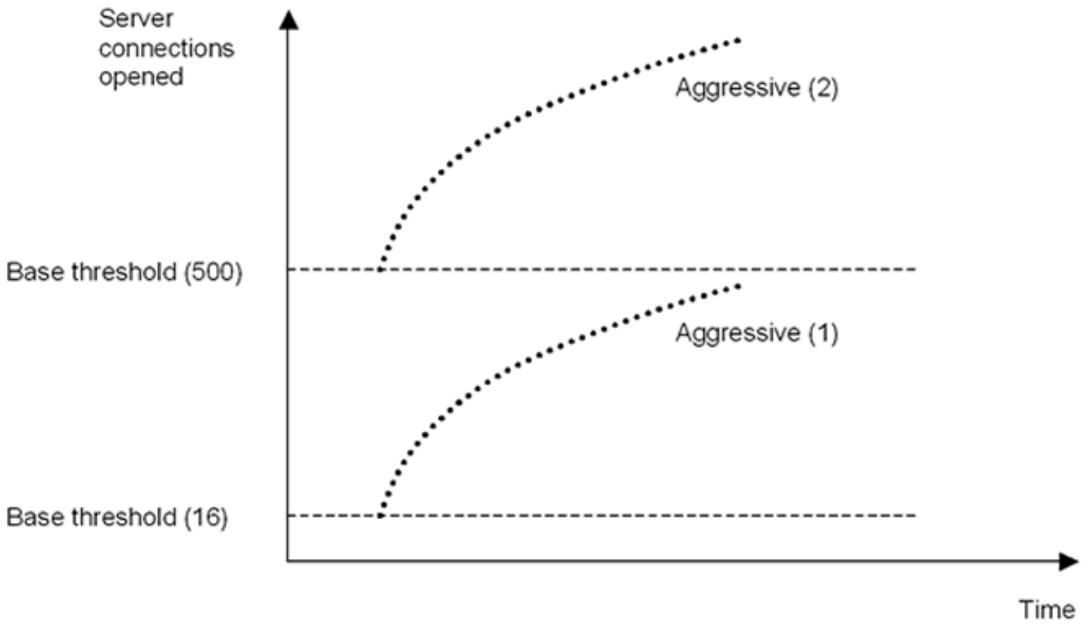


Figure 2. Aggressive Rate with the Default or a Set Base Threshold

To set the threshold for surge protection by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Global System Settings**.
3. If you want to set a base threshold different from the default for the throttle rate, in the **Configure Global Settings** dialog box, **Base Threshold** text box, enter the maximum number of concurrent server connections allowed before surge protection is triggered. The base threshold is the maximum number of server connections that can be open before surge protection is activated. The maximum value for this setting is 32,767 server connections. The default setting for this value is controlled by the throttle rate you choose in the next step.

   Note: If you do not set an explicit value here, the default value will be used.
4. In the **Throttle** drop-down list, select a throttle rate.The throttle is the rate at which the NetScaler appliance allows connections to the server to be opened. The throttle can be set to the following values:

   Aggressive

   Choose this option when the connection-handling and surge-handling capacity of the server is low and the connection needs to be managed carefully. When you set the throttle to aggressive, the base threshold is set to a default value of 16, which means that surge protection is triggered whenever there are 17 or more concurrent

connections to the server.

Normal

Choose this option when there is no external load balancer behind the NetScaler appliance or downstream. The base threshold is set to a value of 200, which means that surge protection is triggered whenever there are 201 or more concurrent connections to the server. Normal is the default throttle option.

Relaxed

Choose this option when the NetScaler appliance is performing load balancing between a large number of Web servers, and can therefore handle a high number of concurrent connections. The base threshold is set to a value of 500, which means that surge protection is triggered only when there are 501 or more concurrent connections to the server.

5. Click **OK**. A message appears in the status bar, stating that the global settings are configured.

Flushing the Surge Queue

When a physical server receives a surge of requests, it becomes slow in responding to the currently connected clients and leaves many users dissatisfied and disgruntled. Often, the overloading also causes the clients to receive error pages. To avoid such overloading, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The NetScaler does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the NetScaler looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the NetScaler establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the NetScaler cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the NetScaler cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The NetScaler maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which NetScaler cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.

Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the NetScaler command line

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the NetScaler flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the NetScaler flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the NetScaler flushes the surge queues of all the entities present on the NetScaler.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the NetScaler command prompt, type:

flush ns surgeQ [**-name** <name>] [**-serverName** <serverName> <port>]

Examples

1.
flush ns surgeQ –name SVC1ANZGB –serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and h
2.
flush ns surgeQ
The above command flushes all the surge queues on the NetScaler.

Parameters for flushing a surge queue

name

Name of a virtual server, service or service group

serverName

Name of a service group member

To flush a surge queue by using the NetScaler configuration utility

1. In the navigation pane, expand **Load Balancing**.
2. To select an entity, do one of the following:
   - To flush the surge queue of a virtual server, click **Virtual Servers**, and then select the virtual server.
   - To flush the surge queue of a service, click **Services**, and then select the service.
   - To flush the surge queue of all the members in a service group, click **Service Groups**, and then select the service group.
   - To flush the surge queue of a specific member in a service group, click **Service Groups**, and in the action pane, click **Manage Members**. In the **Manage Members of a Service Group** dialog box, select the service group member.

   Note: You can select multiple entities in any window.

   Note: To flush the surge queue of a content switching virtual server, in Steps 1 and 2, expand **Content Switching**, and then select a virtual server.
3. In the action pane, click **Flush Surge Queue**.
4. Click **OK**.

   Note: On the NetScaler, if there are other entities with the same name as you selected, you are alerted that the surge queues of those entities would also be flushed. Take an appropriate action.

TCP Buffering

The TCP buffering feature improves the performance of a transaction management environment by adding a speed-matching mechanism between a fast server network and a slow client network and buffering a server's response before delivering it to the client at the client's speed. The server can quickly offload the requested data and then devote its resources to other tasks. Any required retransmission of packets from a server to a client is also done by the Citrix® NetScaler® appliance.

TCP buffering is bypassed for some NetScaler features, including SSL, compression, and caching, because these features perform their own type of buffering. However, TCP buffering is performed for non-compressible and non-cacheable responses from the server, even when compression and caching are enabled. TCP buffering is also skipped for small responses that can fit in a single packet.

You enable or disable the TCP buffering feature globally and on a per-service basis. You can also set the size and memory limit of the buffer.

Note: This content is best understood if you are familiar with creating services and binding them to vservers. For more information, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX128670.

Enabling or Disabling TCP Buffering Globally

TCP buffering is disabled on the appliance by default. When you enable TCP buffering globally, all new services are enabled for TCP buffering by default.

To enable or disable the TCP buffering mode globally by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable mode TCPB
- disable mode TCPB

To enable or disable the TCP buffering mode globally by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Configure modes**.
3. In the **Configure Modes** dialog box, select or clear the **TCP Buffering** check box.
4. Click **OK**.

Enabling or Disabling TCP Buffering Globally

TCP buffering is disabled on the appliance by default. When you enable TCP buffering globally, all new services are enabled for TCP buffering by default.

To enable or disable the TCP buffering mode globally by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands:

- enable mode TCPB
- disable mode TCPB

To enable or disable the TCP buffering mode globally by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under the **Modes and Features** group, click **Configure modes**.
3. In the **Configure Modes** dialog box, select or clear the **TCP Buffering** check box.
4. Click **OK**.

Enabling or Disabling TCP Buffering for a Service

You can enable or disable TCP buffering at the service level. Note that the service level settings take precedence over the global settings.

To enable or disable the TCP buffering mode for a service by using the NetScaler command line

At the NetScaler command prompt, type:

set service <serviceName> **-TCPB** (YES | NO)

To enable or disable the TCP buffering mode for a service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click the service for which you want to enable or disable TCP buffering, and then click **Open**.
3. In the **Configure Service** dialog box, on the **Advanced** tab, under **Settings**, select or clear the **TCP Buffering** check box.
4. Click **OK**.

Setting TCP Buffering Parameters

You can configure two TCP buffering parameters: buffer size and memory usage limit. For best performance, set the connection buffer size so that most responses can fit in the TCP buffer. If integrated caching is not enabled, to provide maximum buffering capacity, increase the memory usage limit to up to half the total system memory.

To set TCP buffering parameters by using the NetScaler command line

At the NetScaler command prompt, type:

set tcpbufParam **-size** <positiveInteger> **-memLimit** <positiveInteger>

Parameters for setting TCP buffering

size

The buffer size is measured in kilobytes and it specifies the size of the TCP buffer per connection. The default size is 64 KB.

memLimit

The memory usage limit parameter specifies the maximum memory that can be used for buffering. The default memory limit is 64 MB.

To set TCP buffering parameters by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Settings**, click **Change TCP parameters**.
3. In the **Configure TCP Parameters** dialog box, under **TCP Buffering**, in the **Buffer Size (KBytes)** text box, type the size of the TCP buffer you want to set, for example, 128.
4. In the **Memory Usage Limit (MBytes)** text box, type the maximum memory size that you want to use for buffering, for example, 128.
5. Click **OK**.

API Documentation

Intended for application developers who want to configure and monitor a NetScaler appliance programmatically.

NITRO API	Describes the use of the NITRO APIs for the REST, Java, and .NET platforms.
XML API	Describes the properties and use of the XML API.

NITRO API

The Citrix® NetScaler® NITRO protocol allows you to configure and monitor the NetScaler appliance programmatically.

NITRO exposes its functionality through Representational State Transfer (REST) interfaces. This ensures that the NITRO functionality can be accessed by applications developed in any programming language. Additionally, for applications that must be developed in Java or .NET, the NITRO protocol is exposed as Java and .NET libraries that are packaged as separate Software Development Kits (SDKs).

Obtaining the Latest NITRO Package

The latest NITRO package is available as a tar file on the **Downloads** page of the NetScaler appliance's configuration utility. You must download and un-tar the file to a folder on your local system. This folder is referred to as <NITRO_SDK_HOME> in this documentation.

The folder contains the NITRO libraries (JARs for Java and DLLs for .NET) in the lib subfolder. The libraries must be added to the client application's classpath to access NITRO functionality. The <NITRO_SDK_HOME> folder also provides samples and documentation that can help you understand the NITRO SDK.

Note: The REST package contains only documentation for using the REST interfaces.

Prerequisites

To use the NITRO protocol, the client application needs only the following:

- Access to a NetScaler appliance, version 9.2 or later.
- A system to generate HTTP or HTTPS requests (payload in JSON format) to the NetScaler appliance. You can use any programming language or a tool to generate the requests.
- For Java clients, you must have a system where Java Development Kit (JDK) 1.5 or later is available. The JDK can be downloaded from http://www.oracle.com/technetwork/java/javase/downloads/index.html.
- For .NET clients, you must have a system with .NET framework 3.5 or later installed. The .NET framework can be downloaded from http://www.microsoft.com/downloads/en/default.aspx.

Note: You must have a basic understanding of the NetScaler appliance before using the NITRO protocol.

NITRO API

The Citrix® NetScaler® NITRO protocol allows you to configure and monitor the NetScaler appliance programmatically.

NITRO exposes its functionality through Representational State Transfer (REST) interfaces. This ensures that the NITRO functionality can be accessed by applications developed in any programming language. Additionally, for applications that must be developed in Java or .NET, the NITRO protocol is exposed as Java and .NET libraries that are packaged as separate Software Development Kits (SDKs).

Obtaining the Latest NITRO Package

The latest NITRO package is available as a tar file on the **Downloads** page of the NetScaler appliance's configuration utility. You must download and un-tar the file to a folder on your local system. This folder is referred to as <NITRO_SDK_HOME> in this documentation.

The folder contains the NITRO libraries (JARs for Java and DLLs for .NET) in the lib subfolder. The libraries must be added to the client application's classpath to access NITRO functionality. The <NITRO_SDK_HOME> folder also provides samples and documentation that can help you understand the NITRO SDK.

Note: The REST package contains only documentation for using the REST interfaces.

Prerequisites

To use the NITRO protocol, the client application needs only the following:

- Access to a NetScaler appliance, version 9.2 or later.
- A system to generate HTTP or HTTPS requests (payload in JSON format) to the NetScaler appliance. You can use any programming language or a tool to generate the requests.
- For Java clients, you must have a system where Java Development Kit (JDK) 1.5 or later is available. The JDK can be downloaded from http://www.oracle.com/technetwork/java/javase/downloads/index.html.
- For .NET clients, you must have a system with .NET framework 3.5 or later installed. The .NET framework can be downloaded from http://www.microsoft.com/downloads/en/default.aspx.

Note: You must have a basic understanding of the NetScaler appliance before using the NITRO protocol.

Execution Flow

The NITRO protocol consists of the client application and the NITRO web service, which runs on the NetScaler appliance. The communication between the client application and the NITRO web service is based on REST architecture using HTTP or HTTPS.

For Java and .NET

NITRO libraries must be integrated with the client application and they act as an interface between the client application and the NITRO web service. Invocation of a NITRO API initiates the following processes:

1. The NITRO library translates the API call into REST request messages that are sent to the NITRO web service.

2. The NITRO web service processes the requests and returns a corresponding REST response message.

3. The NITRO library translates the response into the appropriate response for the API call.

For REST

Invocation of a NITRO REST request initiates the following processes:

1. The application sends REST request messages to the NITRO web service.

2. The NITRO web service processes the requests and returns a corresponding REST response message to the client application.

Execution Flow

The NITRO protocol consists of the client application and the NITRO web service, which runs on the NetScaler appliance. The communication between the client application and the NITRO web service is based on REST architecture using HTTP or HTTPS.

For Java and .NET

NITRO libraries must be integrated with the client application and they act as an interface between the client application and the NITRO web service. Invocation of a NITRO API initiates the following processes:

1. The NITRO library translates the API call into REST request messages that are sent to the NITRO web service.

2. The NITRO web service processes the requests and returns a corresponding REST response message.

3. The NITRO library translates the response into the appropriate response for the API call.

For REST

Invocation of a NITRO REST request initiates the following processes:

1. The application sends REST request messages to the NITRO web service.

2. The NITRO web service processes the requests and returns a corresponding REST response message to the client application.

Tutorials

The tutorials show end-to-end usage of NITRO SDKs (Java and .NET) in developing sample applications.

- Creating Your First NITRO Application

More tutorials will be added in future releases.

Create Your First NITRO Application

After completing this tutorial, you will understand and be able to perform the following tasks:

- Integrate NITRO with the IDE
- Log in to the appliance
- Create a load balancing virtual server (lbvserver)
- Retrieve details of an lbvserver
- Delete an lbvserver
- Save the configurations on the appliance
- Log out of the appliance
- Debug the NITRO application

Before you begin, make sure that you have the latest NITRO SDK and that the client application satisfies the prerequisites for using the NITRO SDK.

Sample Code

For the Java executable code, see the <NITRO_SDK_HOME>/sample/MyFirstNitroApplication.java sample file.

To create your first NITRO application:

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.

2. Create a new class and name it **MyFirstNitroApplication**.

3. Create an instance of `com.citrix.netscaler.nitro.service.nitro_service` class. This instance is used to perform all operations on the appliance:

   nitro_service ns_session = new nitro_service("10.102.29.170","HTTP");

   This code establishes a connection with an appliance that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler appliance that you have access to.
4. Use the `nitro_service` instance to log in to the appliance using your credentials:

   ns_session.login("admin","verysecret");

   This code logs into the appliance, with user name as `admin` and password as `verysecret`. Replace the credentials with your login credentials.
5. Enable the load balancing feature:

   ns_session.enable_features("lb");

   This code first sets the features to be enabled in an array and then enables the LB feature.
6. Create an instance of the `com.citrix.netscaler.nitro.resource.config.lb.lbvserver` class. You will use this instance to perform operations on the lbvserver.

   lbvserver new_lbvserver_obj = new lbvserver();

7. Use the `lbvserver` instance to create a new lbvserver:

   Java

   ```
   new_lbvserver_obj.set_name("MyFirstLbVServer");
   new_lbvserver_obj.set_ipv46("10.102.29.88");
   new_lbvserver_obj.set_port(88);
   new_lbvserver_obj.set_servicetype("HTTP");
   new_lbvserver_obj.set_lbmethod("ROUNDROBIN");
   lbvserver.add(ns_session,new_lbvserver_obj);
   ```

   .NET

   ```
   new_lbvserver_obj.name = "MyFirstLbVServer";
   new_lbvserver_obj.ipv46 = "10.102.29.88";
   new_lbvserver_obj.port = 88;
   new_lbvserver_obj.servicetype = "HTTP";
   new_lbvserver_obj.lbmethod = "ROUNDROBIN";
   lbvserver.add(ns_session,new_lbvserver_obj);
   ```

This code first sets the attributes (name, IP address, service type, and load balancing method) of the lbvserver locally and then adds it to the appliance by using the corresponding `add()` method.

8. Retrieve the details of the lbvserver you have created:

   Java

   ```
   new_lbvserver_obj = lbvserver.get(ns_session,new_lbvserver_obj.get_name());
   System.out.println("Name : " +new_lbvserver_obj.get_name() +"\n" +"Protocol : " +new_lbvserver_obj.ge
   ```

   .NET

   ```
   new_lbvserver_obj = lbvserver.get(ns_session,new_lbvserver_obj.name);
   Console.WriteLine("Name : " +new_lbvserver_obj.name +"\n" +"Protocol : " +new_lbvserver_obj.servicety
   ```

   This code first retrieves the details of the lbvserver as an object from the NetScaler, extracts the required attributes (name and service type) from the object, and displays the results.

9. Delete the lbvserver you created in the above steps:

   Java

   ```
   lbvserver.delete(ns_session, new_lbvserver_obj.get_name());
   ```

   .NET

   ```
   lbvserver.delete(ns_session, new_lbvserver_obj.name);
   ```

10. Save the configurations:

    ```
    ns_session.save_config();
    ```

11. Log out of the appliance:

    ```
    ns_session.logout();
    ```

Debug the NITRO application

All NITRO exceptions are captured by the `com.citrix.netscaler.nitro.exception.nitro_exception` class. For a more detailed description, see Exception Handling.

API Categorization and Usage

The NITRO functionality can be grouped into the following:

- System Management
- Configurations
- Statistics
- AppExpert Application Management
- Exception Handling

Note: All NITRO requests are synchronous. After sending a request, the client application blocks until it receives a response from the NITRO web service.

ns-nitro-manage-sys-new-tsk

Due to technical difficulties, we are unable to display this topic. Citrix is currently fixing this problem. In the meantime, you can view this topic online:

http://support.citrix.com/proddocs/index.jsp?lang=en&topic=/ns-nitro-api-edocs-map/ns-nitro-manage-sys-

ns-nitro-configure-tsk

Due to technical difficulties, we are unable to display this topic. Citrix is currently fixing this problem. In the meantime, you can view this topic online:

http://support.citrix.com/proddocs/index.jsp?lang=en&topic=/ns-nitro-api-edocs-map/ns-nitro-configure-ts

Statistics

The NetScaler appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using the NITRO protocol.

Note: Not all features and resources have statistics objects associated with them.

The following table describes the operations to get NetScaler statistics. For a more detailed description, see the API reference available in the <NITRO_SDK_HOME>/doc folder.

Description	Sample
Java and .NET The APIs to get statistics of a feature are grouped into packages or namespaces that have the format `com.citrix.netscaler.nitro.resource.stat.<feature>`. For example, the statistics of the load balancing feature can be found by using the `com.citrix.netscaler.nitro.resource.stat.lb` package or namespace.	To get statistics of an lbvserver named "MyFirstLbVServer" and print some of the properties returned in the statistics object: Java Sample lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirs System.out.println(stats.get_curclntconnections()); System.out.println(stats.get_deferredreqrate()); .NET Sample lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirs Console.WriteLine(stats.curclntconnections); Console.WriteLine(stats.deferredreqrate);

REST
 	
The URL to get statistics of a feature has the format
`http://<NSIP>/nitro/v1/stat/<feature_name>`
 	
The URL to get the statistics of a resource must have the following format:
`http://<NSIP>/nitro/v1/stat/<resource_type>/<resource_name>`
	To get the statistics of a lbvserver named "MyFirstLbVServer":
	
	• **URL.**
	http://10.102.31.16/nitro/v1/stat/lbvserver/MyFirstLbV
	
	• **HTTP Method.** GET
	
	• **Cookie.** sessionid="##7860...."
	
	• **Response Payload.**
	
	{
	"errorcode": 0,
	"message": "Done"
	"lbvserver":
	[
	{
	"name":"MyFirstLbVServer",
	"establishedconn":0,
	"vslbhealth":0,
	"primaryipaddress":"0.0.0.0",
	...
	}
	]
	}

ns-nitro-manage-appexpert-new-tsk

Due to technical difficulties, we are unable to display this topic. Citrix is currently fixing this problem. In the meantime, you can view this topic online:

http://support.citrix.com/proddocs/index.jsp?lang=en&topic=/ns-nitro-api-edocs-map/ns-nitro-manage-app

Exception Handling

The `errorcode` field indicates the status of the operation.

- An errorcode of 0 indicates that the operation is successful.
- A non-zero errorcode indicates an error in processing the NITRO request.

The error message field provides a brief explanation and the nature of the failure.

Java and .NET: All exceptions in the execution of NITRO APIs are caught by the `com.citrix.netscaler.nitro.exception.nitro_exception` class. To get information about the exception, you can use the `getErrorCode()` method.

REST: The response payload of all operations, specifies the error code and error message. For example, to get the status of a operation:

- **URL.** http://10.102.31.16/nitro/v1/config/snmpoid
- **HTTP Method.** GET
- **Cookie.** sessionid="##78C060..."
- **Response Payload.**

```
{
   "errorcode":1095,
   "message":"Required argument missing ... "
}
```

For a more detailed description of the error codes, see the API reference available in the <NITRO_SDK_HOME>/doc folder.

FAQs

Question: Do all NetScaler features or resources have statistics available or associated with them?

Answer: No. For example, the filter feature does not have a statistics class associated with it, but the DNS feature does.

You can contribute to this section by providing your feedback. For more infromation, see Documentation Feedback.

Unsupported NetScaler Operations

Some NetScaler operations that are available through the GUI and CLI are not available through NITRO APIs. The following list provides the NetScaler operations not supported by NITRO APIs:

- install API
- diff API on nsconfig resource
- UI-internal APIs (update, unset, and get)
- Application firewall APIs:
  - importwsdl
  - importcustom
  - importxsd
  - importxmlerrorpage
  - importhtmlerrorpage
  - rmwsdl
  - rmcustom
  - rmxsd
  - rmxmlerrorpage
  - rmhtmlerrorpage
- CLI-specific APIs:
  - ping
  - ping6
  - traceroute
  - traceroute6
  - nstrace
  - scp
  - configaudit
  - show defaults

- show permission
- batch
- source

XML API

Developers and administrators can use the NetScaler Application Programming Interface (API), nsconfig, to implement customized client applications. The nsconfig API, which mirrors the NetScaler command line interface (CLI), is based on the Web Services Description (WSDL) specification. It includes a filterwsdl command to reduce compilation time and file size. You can secure your API applications at the NetScaler IP address or at the IP address of the subnet on which the NetScaler is deployed.

The following topics describe the properties and use of the API.

Introduction	General information about the API, requirements, and software-version information.
The NS Config Interface	How to use the API.
Examples of API Usage	Basic examples of how to use the API.
The Web Service Description Language (WSDL)	How to use the WSDL-based interface schema to support your client applications, and how to use WSDL Filter to reduce file size and compilation time.
Securing API Access	How to secure API access.

Introduction to the API

The API enables programmatic communications between client applications and the NetScaler appliance, providing the following benefits:

- Developers can control the NetScaler from a custom application. The API enables the client application to configure and monitor the NetScaler.
- Developers can create client applications easily and quickly, using a language and platform with which they are comfortable.
- The API provides a secure, end-to-end, standards-based framework that integrates into the existing infrastructure.

Based on the Simple Object Access Protocol (SOAP) over HTTP, the API consists of the NSConfig interface. NSConfig includes methods for setting and querying the configuration. These methods allow the client application using the NSConfig interface to perform almost all operations that an administrator would normally perform with the CLI or GUI.

In addition, the NetScaler provides an interface description, based on the Web Services Definition Language (WSDL), that facilitates the development of client applications.

Introduction to the API

The API enables programmatic communications between client applications and the NetScaler appliance, providing the following benefits:

- Developers can control the NetScaler from a custom application. The API enables the client application to configure and monitor the NetScaler.
- Developers can create client applications easily and quickly, using a language and platform with which they are comfortable.
- The API provides a secure, end-to-end, standards-based framework that integrates into the existing infrastructure.

Based on the Simple Object Access Protocol (SOAP) over HTTP, the API consists of the NSConfig interface. NSConfig includes methods for setting and querying the configuration. These methods allow the client application using the NSConfig interface to perform almost all operations that an administrator would normally perform with the CLI or GUI.

In addition, the NetScaler provides an interface description, based on the Web Services Definition Language (WSDL), that facilitates the development of client applications.

Hardware and Software Requirements

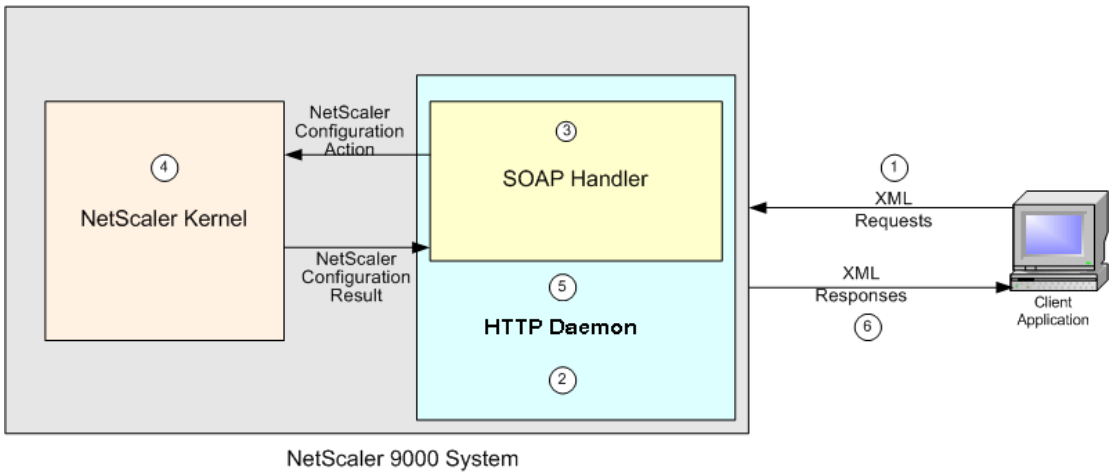
To work with the API, your system needs to meet the following hardware and software setup and requirements:

- A client workstation.
- Access to a NetScaler, version 8.0 or higher.
- A SOAP client tool kit (supporting SOAP version 1.1 and above), and the development environment for the tool kit. For example, if you use a Visual Basic tool kit, you must have Visual Basic installed on your system.

API Architecture

The API architecture is designed to allow NSConfig client requests to be routed, through the HTTP daemon running on the target NetScaler, to a SOAP handler that translates the SOAP request into a call to the (internal) kernel configuration API.

Figure 1. The API Architecture



The order in which the NetScaler processes requests through the API is as follows:

- The client formats a request containing XML conforming to the SOAP protocol and sends it to the NetScaler.
- The HTTPD server instance on the NetScaler routes this request to a SOAP handler.
- The SOAP handler interprets the SOAP headers and maps the enclosed request to an internal configuration function.
- The kernel acts on the request and returns one or more responses.
- The SOAP handler translates the response(s) to a SOAP response message.
- The XML response is sent back to the client in an HTTP response.

The NSConfig Interface

The NSConfig interface closely mirrors the structure of the NetScaler command line interface (CLI). Administrators and programmers who are familiar with the CLI can easily create and implement custom applications to query or set the configuration on their NetScaler.

The NSConfig interface includes methods for most of the CLI commands. In most cases the method and the command name are the same. See the **PortType** section of the WSDL for a complete list of methods and their names.

For example, you use the add lb vserver CLI command to create a load balancing virtual server, as follows:

add lb vserver <vServerName> <serviceType> [<IPAddress> <port>]

where:

<vServerName> = A name for the virtual server.

<serviceType> = (HTTP | FTP | TCP | UDP).

<IPAddress> = The IP address used by the virtual server.

<port> = The port that the virtual server listens on.

Following is the corresponding API call, in the C language:

```
int ns__addlbvserver(void *handle,
      string vServerName,
      string serviceType,
      string IPAddress,
      unsignedShort port,
    ns__addlbvserverResponse *out);
```

Note: The exact syntax of the API call depends on the language used to write the client program. The above `ns__addlbvserver` function prototype is similar to the one that would be generated by the gSOAP package at http://www.cs.fsu.edu/~engelen/soap.html.

The result returned for all NSConfig requests consists of:

Rc

An integer return code. The value is zero if the request succeeded. A non-zero value indicates that the request failed.

Message

A string message. Contains meaningful information only if the request fails (rc is non-zero) (for example, “Required argument missing”).

List

A type-specific list of result entities. This element is present only for requests that retrieve information from the NetScaler. For example, the API method names starting with get, which correspond to the CLI show commands, return a list.

Command names in the NetScaler CLI typically consist of three terms, separated by spaces, identifying the operation, the feature that is being operated on, and the specific item that is being operated on. For example, to create a new Application Firewall profile, you type add appfw profile, followed by the command arguments. The corresponding API methods omit the spaces. For example, the API method for add appfw profile is `addappfwprofile`. The same principle applies to CLI command names that have only two terms. For example, add monitor becomes `addmonitor`. The other exceptions to this pattern are as follows:

1. The CLI show command is changed to get in the API, as shown below.

   show lb vserver => `getlbvserver`

   show service => `getservice`

2. The following commands are omitted from the API:

   - Commands that apply to the CLI itself (for example, clear CLI prompt).
   - The batch, ping, grep, more, shell, and scp commands.
   - The show router bgp and show router map commands.
   - All stat commands.

3. Message "part" names in the API are the same as the corresponding CLI argument names. As in the CLI, case does not matter, and these names can be abbreviated. For more information, see the *Citrix NetScaler Command Reference Guide* at http://support.citrix.com/article/CTX128678.

4. The result of a GET method (which corresponds to a show command in the CLI) is always an array of a type defined in the WSDL. The elements of these complex types generally correspond to arguments to the corresponding add/set command/method.

5. Authorization must be performed once, by sending a login request. The response contains a Set-Cookie HTTP header, and the cookie must be sent with each subsequent request. This is addressed in the Perl examples using by HTTP::Cookies. HTTP::Cookies are used for API client authentication purposes (to log into the NetScaler). In Perl, SOAP::Lite cannot perform this authentication process; HTTP::Cookies are used instead.

6. In some programming languages, such as Perl, it is possible to invoke the programming language API without using the WSDL.

Examples of API Usage

The following examples show how to develop an API call from a standard CLI command, how to generate the SOAP request, and how the NetScaler responds to that request:

Example: Setting the Configuration

Example: Querying the Configuration

Example: Setting the Configuration

This example shows a CLI command, the corresponding API method, the resulting XML request, and the XML response that is sent back to the client.

Note: The actual API method and the XML SOAP message contents may differ from the example shown below. The XML shown will be encased in a SOAP envelope, which will in turn be carried in an HTTP message. For more information, see the W3C web site at http://www.w3.org/TR/SOAP.

The following CLI command creates a Load Balancing virtual server:

> add lb vserver **vipLB1 HTTP 10.100.101.1 80**

Following is the corresponding API method:

> ns__addlbvserver (**handle**, **“vipLB1”**, **“HTTP”**, **“10.100.101.1”**, **80**, **&out**);

The XML generated for this request is as follows.

```
<ns:addlbvserver>
<vServerName xsi:type="xsd:string" >vipLB1</vServerName>
<serviceType xsi:type="ns:vservicetypeEnum>HTTP</ serviceType>
<IPAddress xsi:type="xsd:string">10.100.101.1</IPAddress>
<port xsi:type="xsd:unsignedInt" >80</port>
< /ns:addlbvserver >
```

The XML response to the above request is as follows.

```
<ns:addlbvserverResponse>
<rc xsi:type="xsd:unsignedInt">0</rc>
<message xsi:type="xsd:string">Done</message>
</ns:addlbvserverResponse>
```

Example: Querying the Configuration

This example shows an API request that queries the configuration and receives a list of entities.

Note: The actual API method and the XML SOAP message contents may differ from the example shown below.

The following CLI command shows the configured Load Balancing virtual servers:

> show lb vservers

Sample output of the show lb vservers command is as follows.

```
> show lb vservers
2 configured virtual servers:
1)  vipLB1 (10.100.101.1:80) - HTTP Type: ADDRESS State:
   DOWN
   Method: LEASTCONNECTION Mode: IP
   Persistence: NONE
2)  vipLB2 (10.100.101.2:80) - HTTP Type: ADDRESS State:
   DOWN
   Method: LEASTCONNECTION Mode: IP
   Persistence: NONE
Done
```

Following is the corresponding API method to show the list of Load Balancing virtual servers.

ns__getlbvserver(**handle**, **NULL**, **&out**)

The XML generated for this request is as follows.

```
<ns:getlbvserver></ns:getlbvserver>
```

The XML response to the above request is as follows.

```
<ns:getlbvserverResponse>
   <rc xsi:type="xsd:unsignedInt">0</rc>
   <message xsi:type="xsd:string">Done</message>
   <List xsi:type="SOAP-ENC:Array"
      SOAP-ENC:arrayType="ns:lbvserver[2]">
      <item xsi:type="ns:lbvserver">
      <vServerName xsi:type="xsd:string>vipLB1
         </vServerName>
         <serviceType xsi:type="xsd:string>HTTP</ serviceType>
         <IPAddress xsi:type="xsd:string >10.100.101.1
         </IPAddress>
      <port xsi:type="xsd:unsignedInt">80</port>
   </item>
   <item xsi:type="ns:lbvserver">
```

```
            <vServerName xsi:type="xsd:string>vipLB2
            </vServerName>
            <serviceType xsi:type="xsd:string>HTTP</ serviceType>
            <IPAddress xsi:type="xsd:string >10.100.101.2
            </IPAddress>
               <port xsi:type="xsd:unsignedInt">80</port>
            </item>
         </List>
      </ns:getlbvserverResponse>
```

The Web Service Definition Language (WSDL)

The NetScaler WSDL describes services for the entire range of NetScaler services. The NetScaler provides two WSDL files:

NSConfig.wsdl

Configuration APIs are defined in this file. The NSConfig.wsdl file is found on the NetScaler at http://<NSIP>/api/NSConfig.wsdl, where <NSIP> is the IP address of your NetScaler. This file is much larger than the NSStat.wsdl file. With the help of a third-party tool (such as gSOAP), developers can use this file to generate client stubs. A custom application can then call the stubs to send requests to the NetScaler. The application can be in any standard programming language that is supported by the third-party tool. Common programming languages for this purpose include Perl, Java, C, and C#. You can use the filterwsdl command to select only the service definitions that are relevant to the API calls made in your script.

NSStat.wsdl

Statistical APIs are defined in this file. The NSStat.wsdl file is found on the NetScaler at http://<NSIP>/api/NSStat.wsdl, where <NSIP> is the IP address of your NetScaler.

Creating Client Applications with the NSConfig.wsdl File

A client application can be created by importing the NSConfig.wsdl file with the gSOAP WSDL Importer to create a header file with C or C++ declarations of the SOAP methods. The gSOAP compiler is then used to translate this header file into stubs for the client application.

1. Get the NSConfig.h header file from the WSDL file.

    a. Run the wsdl2h program that comes with gSOAP on the WSDL file. The wsdl2h program is in the following location.

    > ./wsdl2h NSConfig.wsdl

    The output of wsdl2h is as follows:

    ```
    **  The gSOAP WSDL parser for C and C++ 1.0.2
    **  Copyright (C) 2001-2004 Robert van Engelen, Genivia, Inc.
    **  All Rights Reserved. This product is provided "as is", without any warranty.
    Saving NSConfig.h
    Reading file 'NSConfig.wsdl'
    Cannot open file 'typemap.dat'
    Problem reading type map file typemap.dat.
    Using internal type definitions for C instead.
    ```

    b. Run the soapcpp2 program to compile the header file and complete the process, as shown below. > soapcpp2 NSConfig.h
2. Generate the XML files and stubs as follows:

    > ./soapcpp2 -c -i NSConfig.h

    Following is sample output for this command:

    ```
    **  The gSOAP Stub and Skeleton Compiler for C and C++ 2.4.1
    **  Copyright (C) 2001-2004 Robert van Engelen, Genivia, Inc.
    **  All Rights Reserved. This product is provided "as is", without any warranty.
    Saving soapStub.h
    Saving soapH.h
    Saving soapC.c
    Saving soapClient.c
    Saving soapServer.c
    Saving soapClientLib.c
    Saving soapServerLib.c
    Using ns1 service name: NSConfigBinding
    Using ns1 service location: http://NetScaler.com/api Using ns1 schema namespace: urn:NSConfig
    Saving soapNSConfigBindingProxy.h client proxy
    Saving soapNSConfigBindingObject.h server object
    ```

```
Saving NSConfigBinding.addserver.req.xml sample SOAP/XML request
Saving NSConfigBinding.addserver.res.xml sample SOAP/XML response
Saving NSConfigBinding.disableserver.req.xml sample SOAP/ XML request
Saving NSConfigBinding.disableserver.res.xml sample SOAP/ XML response
Saving NSConfigBinding.enableserver.req.xml sample SOAP/ XML request
Saving NSConfigBinding.enableserver.res.xml sample SOAP/ XML response
[ ... Similar lines clipped ... ]
Saving NSConfigBinding.nsmap namespace mapping table
Compilation successful
```

This creates the stub files soapC.c, soapClient.c and stdsoap2.c.

3. Link the stub files you created with your source code to create a stand-alone binary that invokes the API.

Filter WSDL

The NetScaler WSDL describes services for the entire range of NetScaler services. When you use the NetScaler API in your scripts, by linking to the WSDL and attempting to compile the application, the entire WSDL is included, unnecessarily increasing compilation time and the size of the program.

Filter WSDL is a tool for selecting only those service definitions from the NetScaler WSDL that are relevant to the API calls made in the script. You can use the filter WSDL tool to filter NSConfig.wsdl and NSStat.wsdl files.

The NetScaler provides two WSDL files, one for the configuration APIs (NSConfig.wsdl) and the other for statistical APIs (NSStat.wsdl). The WSDL file for the configuration API is much larger. Therefore, it is important to use filter WSDL when compiling programs written with the configuration API.

Filter WSDL is a program that works on the Windows, FreeBSD and Linux platforms, and it can be run from the CLI.

The syntax for running filter WSDL is as follows:

filterwsdl <fromwsdl> <pattern>

where:

fromwsdl = The wsdl file that you want to filter

pattern = API method names or patterns that should be filtered

For example, if you want to filter all the service definitions for the API method addlbvserver from the NetScaler WSDL file, NSConfig.wsdl, you can use the command:

> filterwsdl NSConfig.wsdl "addlbvserver"

The output of this command is sent to the screen by default, but it can be redirected to a file on the NetScaler by using the UNIX redirect operator (>). The output of the previous command can be saved into a file called NSConfig-Custom.wsdl by using the command as follows:

> filterwsdl NSConfig.wsdl "addlbvserver" > NSConfig-Custom.wsdl

In this case, the original WSDL file is 1.58 MB, but the filtered WSDL file is 6 KB.

The pattern used in the filterwsdl command can include the + and - operators and the wildcard operator (*) to create more generic filters.

For example, if you want to filter the service definitions for all the available load balancing methods, you can use the following command:

> filterwsdl NSConfig.wsdl "*lb"*

This command will filter all the Load Balancing methods but will also include GSLB methods, because the pattern lb will be matched by all GSLB methods also. To include only LB methods and exclude all GSLB methods, use the command as follows:

> filterwsdl NSConfig.wsdl +"*lb" -"glsb"*

Securing API Access

Secure access to CLI objects can be based on the NetScaler IP address or on the subnet IP address on which the NetScaler is deployed. To provide secured API access based on the NetScaler IP address, you must configure the NetScaler to use transparent SSL mode with clear text port.

To configure secure API access based on the NetScaler IP

1. Create a loopback SSL service and configure it use transparent SSL mode with clear text port:

   add service secure_xmlaccess 127.0.0.1 **SSL** 443 **-clearTextPort** 80

2. Add certificate and key:

   add certkey cert1 **-cert** /nsconfig/ssl/ssl/cert1024.pem **-key** /nsconfig/ssl/ssl/rsakey.pem

   Note: You can use an existing certificate and key or use the NetScaler Certificate Authority Tool to create a key and test certificate for secure access.

3. Bind the certificate and key to the service:

   bind certkey secure_xmlaccess cert1 **-Service**

4. Add a custom TCP monitor to monitor the SSL service you have added:

   add monitor ssl_mon TCP **-destport** 80

5. Bind the custom TCP monitor to the SSL service:

   bind monitor ssl_mon secure_xmlaccess

To configure secure API access based on the subnet IP

1. Create an SSL VIP in the appropriate subnet:

   add vserver <vServerName> SSL <Subnet-IP> 443

2. Create a loopback HTTP service:

   add service <serviceName> 127.0.0.1 HTTP 80

3. Bind the service to the SSL VIP:

   bind lb vserver <vServerName> <serviceName>

4. Add the certificate and the key:

   add certkey cert1 **-cert** /nsconfig/ssl/ssl/cert1024.pem **-key** /nsconfig/ssl/ssl/rsakey.pem

   Note: You can use an existing certificate and key or use the NetScaler Certificate Authority Tool to create a key and test certificate.

5. Bind the Certificate and the Key to the SSL VIP:

   bind certkey <vServerName> cert1

DataStream Reference

This appendix describes the MySQL protocol and the character sets supported by the DataStream feature. It also describes how NetScaler handles transaction requests and special queries that modify the state of a connection.

Supported Protocols and Database Versions

The NetScaler DataStream feature supports MySQL protocol version 4.1.

For information about the MySQL protocol, see http://forge.mysql.com/wiki/MySQL_Internals_ClientServer_Protocol.

The MySQL database versions supported are 4.1 and 5.0.

The NetScaler DataStream feature supports TDS protocol version 7.1 and higher for MS SQL databases.

For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

The MS SQL database versions supported are 2005, 2008, and 2008 R2.

Supported Protocols and Database Versions

The NetScaler DataStream feature supports MySQL protocol version 4.1.

For information about the MySQL protocol, see http://forge.mysql.com/wiki/MySQL_Internals_ClientServer_Protocol.

The MySQL database versions supported are 4.1 and 5.0.

The NetScaler DataStream feature supports TDS protocol version 7.1 and higher for MS SQL databases.

For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

The MS SQL database versions supported are 2005, 2008, and 2008 R2.

Character Sets

The DataStream feature supports only the UTF-8 charset.

The character set used by the client while sending a request may be different from the character set used in the database server responses. Although the charset parameter is set during the connection establishment, it can be changed at any time by sending an SQL query. The character set is associated with a connection, and therefore, requests on connections with one character set cannot be multiplexed onto a connection with a different character set.

NetScaler parses the queries sent by the client and the responses sent by the database server.

The character set associated with a connection can be changed after the initial handshake by using the following two queries:

- **SET NAMES** <charset> **COLLATION** <collation>
- **SET CHARACTER SET** <charset>

Transactions

In MySQL, transactions are identified by using the connection parameter AUTOCOMMIT or the BEGIN:COMMIT queries. The AUTOCOMMIT parameter can be set during the initial handshake, or after the connection is established by using the query SET AUTOCOMMIT.

NetScaler explicitly parses each and every query to determine the beginning and end of a transaction.

In MySQL protocol, the response contains two flags to indicate whether the connection is a transaction, the TRANSACTION and AUTOCOMMIT flags.

If the connection is a transaction, the TRANSACTION flag is set. Or, if the AutoCommit mode is OFF, the AUTOCOMMIT flag is not set. NetScaler parses the response, and if either the TRANSACTION flag is set or the AUTOCOMMIT flag is not set, it does not do connection multiplexing. When these conditions are no longer true, the NetScaler begins connection multiplexing.

Special Queries

There are special queries, such as SET and PREPARE, that modify the state of the connection and may break request switching, and therefore, these need to be handled differently.

On receiving a request with special queries, NetScaler sends an OK response to the client and additionally, stores the request in the connection.

When a non-special query, such as INSERT and SELECT, is received along with a stored query, the NetScaler first, looks for the server-side connection on which the stored query has already been sent to the database server. If no such connections exist, NetScaler creates a new connection, and sends the stored query first, and then, sends the request with the non-special query.

As of now, only 16 queries are stored in each connection.

The following is a list of the special queries for which NetScaler has a modified behavior.

SET query

The SET SQL queries define variables that are associated with the connection. These queries are also used to define global variables, but as of now, NetScaler is unable to differentiate between local and global variables. For this query, the NetScaler uses the 'store and forward' mechanism described earlier in this section.

USE <db> query

Using this query, the user can change the database associated with a connection. In this case, NetScaler parses the <db> value sent. For this query, the NetScaler uses the 'store and forward' mechanism described earlier in this section.

INIT_DB command

For this command, NetScaler stops further request switching.

COM_PREPARE

NetScaler stops request switching on receiving this command.

PREPARE query

This query is used to create prepared statements that are associated with a connection. For this query, the NetScaler uses the 'store and forward' mechanism described earlier in this section.

Policy Configuration and Reference

The following topics provide the conceptual and reference information that you require for configuring advanced policies on the Citrix® NetScaler® appliance.

Introduction to Policies and Expressions	Describes the purpose of expressions, policies, and actions, and how different NetScaler applications make use of them.
Configuring Advanced Policies	Describes the structure of advanced policies and how to configure them individually and as policy banks.
Configuring Advanced Expressions: Getting Started	Describes expression syntax and semantics, and briefly introduces how to configure expressions and policies.
Advanced Expressions: Evaluating Text	Describes expressions that you configure when you want to operate on text (for example, the body of an HTTP POST request or the contents of a user certificate).
Advanced Expressions: Working with Dates, Times, and Numbers	Describes expressions that you configure when you want to operate on any type of numeric data (for example, the length of a URL, a client's IP address, or the date and time that an HTTP request was sent).
Advanced Expressions: Parsing HTTP, TCP, and UDP Data	Describes expressions for parsing IP and IPv6 addresses, MAC addresses, and data that is specific to HTTP and TCP traffic.
Advanced Expressions: Parsing SSL Certificates	Describes how to configure expressions for SSL traffic and client certificates, for example, how to retrieve the expiration date of a certificate or the certificate issuer.
Advanced Expressions: IP and MAC Addresses, Throughput, VLAN IDs	Describes expressions that you can use to work with any other client- or server-related data not discussed in other chapters.
Typecasting Data	Describes expressions for transforming data of one type to another.
Regular Expressions	Describes how to pass regular expressions as arguments to operators in advanced expressions.
Configuring Classic Policies and Expressions	Provides details on how to configure the simpler policies and expressions known as classic policies and classic expressions.

Expressions Reference	A reference for classic and advanced expression arguments.
Summary Examples of Advanced Expressions and Policies	Examples of classic and advanced expressions and policies, in both quick reference and tutorial format, that you can customize for your own use.
Tutorial Examples of Advanced Policies for Rewrite	Examples of advanced policies for use in the Rewrite feature.
Tutorial Examples of Classic Policies	Examples of classic policies for NetScaler features such as Application Firewall and SSL.
Migration of Apache mod_rewrite Rules to Advanced Policies	Examples of functions that were written using the Apache HTTP Server mod_rewrite engine, with examples of these functions after translation into Rewrite and Responder policies on the NetScaler.

Introduction to Policies and Expressions

For many NetScaler features, policies control how a feature evaluates data, which ultimately determines what the feature does with the data. A policy uses a logical expression, also called a rule, to evaluate requests, responses, or other data, and applies one or more actions determined by the outcome of the evaluation. Alternatively, a policy can apply a profile, which defines a complex action.

Some NetScaler features use default syntax policies, which provide greater capabilities than do the older, classic, policies. If you migrated to a newer release of the NetScaler software and have configured classic policies for features that now use default syntax policies, you might have to manually migrate policies to the default syntax.

Introduction to Policies and Expressions

For many NetScaler features, policies control how a feature evaluates data, which ultimately determines what the feature does with the data. A policy uses a logical expression, also called a rule, to evaluate requests, responses, or other data, and applies one or more actions determined by the outcome of the evaluation. Alternatively, a policy can apply a profile, which defines a complex action.

Some NetScaler features use default syntax policies, which provide greater capabilities than do the older, classic, policies. If you migrated to a newer release of the NetScaler software and have configured classic policies for features that now use default syntax policies, you might have to manually migrate policies to the default syntax.

Classic and Default Syntax Policies

Classic policies evaluate basic characteristics of traffic and other data. For example, classic policies can identify whether an HTTP request or response contains a particular type of header or URL.

Default syntax policies can perform the same type of evaluations as classic policies. In addition, default syntax policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header).

In addition to assigning a policy an action or profile, you bind the policy to a particular point in the processing associated with the NetScaler features. The bind point is one factor that determines when the policy will be evaluated.

Benefits of Using Default Syntax Policies

Default syntax policies use a powerful expression language that is built on a class-object model, and they offer several options that enhance your ability to configure the behavior of various NetScaler features. With default syntax policies, you can do the following:

- Perform fine-grained analyses of network traffic from layers 2 through 7.
- Evaluate any part of the header or body of an HTTP or HTTPS request or response.
- Bind policies to the multiple bind points that the default syntax policy infrastructure supports at the default, override, and virtual server levels.
- Use goto expressions to transfer control to other policies and bind points, as determined by the result of expression evaluation.
- Use special tools such as pattern sets, policy labels, rate limit identifiers, and HTTP callouts, which enable you to configure policies effectively for complex use cases.

Additionally, the configuration utility extends robust graphical user interface support for default syntax policies and expressions and enables users who have limited knowledge of networking protocols to configure policies quickly and easily. The configuration utility also includes a policy evaluation feature for default syntax policies. You can use this feature to evaluate a default syntax policy and test its behavior before you commit it, thus reducing the risk of configuration errors.

Basic Components of a Classic or Default Syntax Policy

Following are a few characteristics of both classic and default syntax policies:

Name.

Each policy has a unique name.

Rule.

The rule is a logical expression that enables the NetScaler feature to evaluate a piece of traffic or another object.

For example, a rule can enable the NetScaler to determine whether an HTTP request originated from a particular IP address, or whether a Cache-Control header in an HTTP request has the value “No-Cache.”

Default syntax policies can use all of the expressions that are available in a classic policy, with the exception of classic expressions for the SSL VPN client. In addition, default syntax policies enable you to configure more complex expressions.

Bindings.

To ensure that the NetScaler can invoke a policy when it is needed, you associate the policy, or bind it, to one or more bind points.

You can bind a policy globally or to a virtual server. For more information, see About Policy Bindings.

An associated action.

An action is a separate entity from a policy. Policy evaluation ultimately results in the NetScaler performing an action.

For example, a policy in the integrated cache can identify HTTP requests for .gif or .jpeg files. An action that you associate with this policy determines that the responses to these types of requests are served from the cache.

For some features, you configure actions as part of a more complex set of instructions known as a profile. For more information, see Order of Evaluation Based on Traffic Flow.

How Different NetScaler Features Use Policies

The NetScaler supports a variety of features that rely on policies for operation. The following table summarizes how the NetScaler features use policies.

Table 1. NetScaler Feature, Policy Type, and Policy Usage

Feature Name	Policy Type	How You Use Policies in the Feature
System	Classic	For the Authentication function, policies contain authentication schemes for different authentication methods. For example, you can configure LDAP and certificate-based authentication schemes. You also configure policies in the Auditing function.
DNS	Default	To determine how to perform DNS resolution for requests.

SSL	Classic	To determine when to apply an encryption function and add certificate information to clear text. To provide end-to-end security, after a message is decrypted, the SSL feature re-encrypts clear text and uses SSL to communicate with Web servers.
Compression	Classic and Default	To determine what type of traffic is compressed.
Integrated Caching	Default	To determine whether HTTP responses are cacheable.
Responder	Default	To configure the behavior of the Responder function.
Protection Features	Classic	To configure the behavior of the Filter, SureConnect, and Priority Queueing functions.
Content Switching	Classic and Default	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type, and associated cache server.

AAA - Traffic Management	Classic Exceptions: • Traffic policies support only default syntax policies • Authorization policies support both classic and default syntax policies.	To check for client-side security before users log in and establish a session. Traffic policies, which determine whether single sign-on (SSO) is required, use only the default syntax. Authorization policies authorize users and groups that access intranet resources through the appliance.
Cache Redirection	Classic	To determine whether responses are served from a cache or from an origin server.

Rewrite	Default	To identify HTTP data that you want to modify before serving. The policies provide rules for modifying the data. For example, you can modify HTTP data to redirect a request to a new home page, or a new server, or a selected server based on the address of the incoming request, or you can modify the data to mask server information in a response for security purposes. The URL Transformer function identifies URLs in HTTP transactions and text files for the purpose of evaluating whether a URL should be transformed.
Application Firewall	Classic and Default	To identify characteristics of traffic and data that should or should not be admitted through the firewall.
Access Gateway, Clientless Access function	Default	To define rewrite rules for general Web access using the Access Gateway.
Access Gateway	Classic	To determine how the Access Gateway performs authentication, authorization, auditing, and other functions.

About Actions and Profiles

Policies do not themselves take action on data. Policies provide read-only logic for evaluating traffic. To enable a feature to perform an operation based on a policy evaluation, you configure actions or profiles and associate them with policies.

Note: Actions and profiles are specific to particular features. For information about assigning actions and profiles to features, see the documentation for the individual features.

About Actions

Actions are steps that the NetScaler takes, depending on the evaluation of the expression in the policy. For example, if an expression in a policy matches a particular source IP address in a request, the action that is associated with this policy determines whether the connection is permitted.

The types of actions that the NetScaler can take are feature specific. For example, in Rewrite, actions can replace text in a request, change the destination URL for a request, and so on. In Integrated Caching, actions determine whether HTTP responses are served from the cache or an origin server.

In some NetScaler features actions are predefined, and in others they are configurable. In some cases, (for example, Rewrite), you configure the actions using the same types of expressions that you use to configure the associated policy rule.

About Profiles

Some NetScaler features enable you to associate profiles, or both actions and profiles, with a policy. A profile is a collection of settings that enable the feature to perform a complex function. For example, in the Application Firewall, a profile for XML data can perform multiple screening operations, such as examining the data for illegal XML syntax or evidence of SQL injection.

Use of Actions and Profiles in Particular Features

The following table summarizes the use of actions and profiles in different NetScaler features. The table is not exhaustive. For more information about specific uses of actions and profiles for a feature, see the documentation for the feature.

Table 1. Use of Actions and Profiles in Different NetScaler Features

re	Use of an Action	Use of a Profile

cation all	Synonymous with a profile	All Application Firewall features use profiles to defi behaviors, including pattern-based learning. You add these profiles to policies.
s vay	The following features of the Access Gateway use actions: • **Pre-Authentication.** Uses Allow and Deny actions. You add these actions to a profile. • **Authorization.** Uses Allow and Deny actions. You add these actions to a policy. • **TCP Compression.** Uses various actions. You add these actions to a policy.	The following features use a profile: • Pre-Authentication • Session • Traffic • Clientless Access After configuring the profiles, you add them to polic
te	You configure URL rewrite actions and add them to a policy.	Not used.
ated ng	You configure caching and invalidation actions within a policy	Not used.
Traffic gement	You select an authentication type, set an authorization action of ALLOW or DENY, or set auditing to SYSLOG or NSLOG.	You can configure session profiles with a default tim authorization action.
ction res	You configure actions within policies for the following functions: • Filter • Compression • Responder • SureConnect	Not used.
	You configure actions within SSL policies	Not used.
m	The action is implied. For the Authentication function, it is either Allow or Deny. For Auditing, it is Auditing On or Auditing Off.	Not used.
	The action is implied. It is either Drop Packets or the location of a DNS server.	Not used.
ffload	The action is implied. It is based on a policy that you associate with an SSL virtual server or a service.	Not used.
ression	Determine the type of compression to apply to the data	Not used.
nt ning	The action is implied. If a request matches the policy, the request is directed to the virtual server associated with the policy.	Not used.
ection	The action is implied. If a request matches the policy, the request is directed to the origin server.	Not used.

About Policy Bindings

A policy is associated with, or bound to, an entity that enables the policy to be invoked. For example, you can bind a policy to request-time evaluation that applies to all virtual servers. A collection of policies that are bound to a particular bind point constitutes a policy bank.

Following is an overview of different types of bind points for a policy:

Request time global.

A policy can be available to all components in a feature at request time.

Response time global.

A policy can be available to all components in a feature at response time.

Request time, virtual server-specific.

A policy can be bound to request-time processing for a particular virtual server. For example, you can bind a request-time policy to a cache redirection virtual server to ensure that particular requests are forwarded to a load balancing virtual server for the cache, and other requests are sent to a load balancing virtual server for the origin.

Response time, virtual server-specific.

A policy can also be bound to response-time processing for a particular virtual server.

User-defined policy label.

For default syntax policies, you can configure custom groupings of policies (policy banks) by defining a policy label and collecting a set of related policies under the policy label.

Other bind points.

The availability of additional bind points depends on type of policy (classic or default syntax), and specifics of the relevant NetScaler feature. For example, classic policies that you configure for the Access Gateway have user and group bind points.

For additional information about default syntax policy bindings, see Binding Policies That Use the Default Syntax and Configuring a Policy Bank for a Virtual Server. For additional information about classic policy bindings, see Configuring a Classic Policy.

About Evaluation Order of Policies

For classic policies, policy groups and policies within a group are evaluated in a particular order, depending on the following:

- The bind point for the policy, for example, whether the policy is bound to request-time processing for a virtual server or global response-time processing. For example, at request time, the NetScaler evaluates all request-time classic policies before evaluating any virtual server-specific policies.

- The priority level for the policy. For each point in the evaluation process, a priority level that is assigned to a policy determines the order of evaluation relative to other policies that share the same bind point. For example, when the NetScaler evaluates a bank of request-time, virtual server-specific policies, it starts with the policy that is assigned to the lowest priority value. In classic policies, priority levels must be unique across all bind points.

For default syntax policies, as with classic policies, the NetScaler selects a grouping, or bank, of policies at a particular point in overall processing. Following is the order of evaluation of the basic groupings, or banks, of default syntax policies:

1. Request-time global override

2. Request-time, virtual server-specific (one bind point per virtual server)

3. Request-time global default

4. Response-time global override

5. Response-time virtual server-specific

6. Response-time global default

However, within any of the preceding banks of policies, the order of evaluation is more flexible than in classic policies. Within a policy bank, you can point to the next policy to be evaluated regardless of the priority level, and you can invoke policy banks that belong to other bind points and user-defined policy banks.

Order of Evaluation Based on Traffic Flow

As traffic flows through the NetScaler and is processed by various features, each feature performs policy evaluation. Whenever a policy matches the traffic, the NetScaler stores the action and continues processing until the data is about to leave the NetScaler. At that point, the NetScaler typically applies all matching actions. Integrated Caching, which only applies a final Cache or NoCache action, is an exception.

Some policies affect the outcome of other policies. Following are examples:

- If a response is served from the integrated cache, some other NetScaler features do not process the response or the request that initiated it.
- If the Content Filtering feature prevents a response from being served, no subsequent features evaluate the response.

If the Application Firewall rejects an incoming request, no other features can process it.

Classic and Default Syntax Expressions

One of the most fundamental components of a policy is its rule. A policy rule is a logical expression that enables the policy to analyze traffic. Most of the policy's functionality is derived from its expression.

An expression matches characteristics of traffic or other data with one or more parameters and values. For example, an expression can enable the NetScaler to accomplish the following:

- Determine whether a request contains a certificate.
- Determine the IP address of a client that sent a TCP request.
- Identify the data that an HTTP request contains (for example, a popular spreadsheet or word processing application).
- Calculate the length of an HTTP request.

About Classic Expressions

Classic expressions enable you to evaluate basic characteristics of data. They have a structured syntax that performs string matching and other operations.

Following are a few simple examples of classic expressions:

- An HTTP response contains a particular type of Cache Control header.

```
res.http.header Cache-Control contains public
```

- An HTTP response contains image data.

```
res.http.header Content-Type contains image/
```

- An SSL request contains a certificate.

```
req.ssl.client.cert exists
```

About Default Syntax Expressions

Any feature that uses default syntax policies also uses default syntax expressions. For information about which features use default syntax policies, see the table NetScaler Feature, Policy Type, and Policy Usage.

Default syntax expressions have a few other uses. In addition to configuring default syntax expressions in policy rules, you configure default syntax expressions in the following situations:

Integrated Caching:

You use default syntax expressions to configure a selector for a content group in the integrated cache.

Load Balancing:

You use default syntax expressions to configure token extraction for a load balancing virtual server that uses the TOKEN method for load balancing.

Rewrite:

You use default syntax expressions to configure rewrite actions.

Rate-based policies:

You use default syntax expressions to configure limit selectors when configuring a policy to control the rate of traffic to various servers.

Following are a few simple examples of default syntax expressions:

- An HTTP request URL contains no more than 500 characters.

```
http.req.url.length <= 500
```

- An HTTP request contains a cookie that has fewer than 500 characters.

```
http.req.cookie.length < 500
```

- An HTTP request URL contains a particular text string.

```
http.req.url.contains(".html")
```

Converting Classic Expressions to the Newer Default Expression Syntax

You can convert a classic expression to the default expression syntax by using the nspepi conversion tool. You can also use the tool to convert all the classic expressions in the NetScaler configuration to the default syntax (with the exception of NetScaler entities that currently support only classic expressions).

The conversion tool does not convert policies configured for the following features, because the features currently support only classic policies:

- Authentication, Pre-authentication
- SSL
- Cache redirection
- VPN (session, traffic, and tunnel traffic)
- Content filtering (The responder feature not only provides you with functionality that is equivalent to that provided by the content filtering feature but also surpasses the content filtering feature in the use cases that it supports. Additionally, responder supports the more powerful default syntax for policy expressions.)

The following NetScaler features support both classic and default syntax expressions and, therefore, support the conversion of classic expressions to default syntax expressions:

- Application Firewall policies
- Authorization policies
- Named expressions
- Compression policies
- Content switching policies
- User-defined, rule-based tokens/persistency (the –rule parameter value that is specified for a load balancing virtual server)

About the Conversion Process

When parsing a NetScaler configuration file, the conversion tool performs the following actions:

1. In commands that create classic named expressions, the conversion tool replaces the names of the classic expressions with default syntax expressions.

2. In commands that support only the classic syntax, if classic named expressions are used, the conversion tool replaces the names of the classic expressions with the actual classic expressions they represent. This action ensures that the names of expressions in classic-only features do not reference the default syntax expressions created from Step 1.

3. In commands associated with entities that support both the classic syntax and the default syntax, the conversion tool replaces all classic expressions in commands with default syntax expressions.

Example

Consider the following sample configuration commands:

```
add policy expression ne_c1 "METHOD == GET"
add policy expression ne_c2 "ne_c1 || URL == /*.htm "
add filter policy pol1 -rule "ne_c2" -reqAction YES
add cmp policy pol2 -rule "REQ.HTTP.HEADER Accept CONTAINS \'text/html\'" -resAction COMPRESS
add cmp policy pol3 -rule "ne_c1 || ne_c2" -resAction GZIP
```

In the commands that create the classic named expressions `ne_c1` and `ne_c2`, the tool replaces the names of the expressions with actual default syntax expressions. This action, which corresponds to Step 1 described earlier, results in the following commands:

```
add policy expression ne_c1 "HTTP.REQ.METHOD.EQ(\"GET\")"

add policy expression ne_c2 "HTTP.REQ.URL.SUFFIX.EQ(\"htm\")"
```

The filter policy command supports only the classic syntax. Therefore, the conversion tool replaces the classic named expression `ne_c1` with the actual classic expression it represents. Note that the tool replaces `ne_c1` in the expression for `ne_c2`, and then replaces `ne_c2` in the filter policy with the classic expression. This action, which corresponds to Step 2 described earlier, results in the following command:

```
add filter policy pol1 -rule "METHOD == GET || URL == /*.htm"
-reqAction YES
```

The compression feature supports both classic and default syntax expressions. Therefore, in the command that creates the compression policy `pol2`, the conversion tool replaces the expression with a default syntax expression. This action, which corresponds to Step 3 described earlier, results in the following command:

```
add cmp policy pol2 -rule
"HTTP.REQ.HEADER(\"Accept\").AFTER_STR(\"text/html\").LENGTH.GT(0)"
-resAction COMPRESS
```

The command that creates the compression policy `pol3` is unaffected by the conversion process because, after the conversion process is complete, `ne_c1` and `ne_c2` reference the default syntax expressions that result from Step 1.

Client security messages are not supported in the newer default policy format and, therefore, are lost. The `SYS.EVAL_CLASSIC_EXPR` function is replaced with a default policy expression. The following entities support the `SYS.EVAL_CLASSIC_EXPR` function:

- DNS policies
- Rate limit selectors
- Cache selectors
- Cache policies
- Content switching policies
- Rewrite policies
- URL transformation policies
- Responder policies
- Application Firewall policies
- Authorization policies
- Compression policies
- CVPN access policies

After performing the conversion, the tool saves the changes in a new configuration file. The new configuration file is created in the directory in which the input file exists. The name of the new configuration file is the same as the name of the input configuration file except for the string `new_` used as a prefix. Conversion warnings are reported in a warning line at the end of the screen output. Additionally, a warning file is created in the directory in which the input configuration file resides. For more information about the warning file and the types of warnings that are reported, see Conversion Warnings.

Converting Expressions

You can use the `nspepi` tool to convert a single classic expression to the default syntax. The `nspepi` tool must be run from the shell prompt on the NetScaler appliance.

To convert a classic expression to the default syntax by using the NetScaler command line interface

At the shell prompt, type:

```
nspepi -e "<classic expression>"
```

Example

```
root@NS# nspepi -e "REQ.HTTP.URL == /*.htm"
 "HTTP.REQ.URL.REGEX_MATCH(re#/(.*)\.htm#)"
```

Parameters for converting a classic expression to a default syntax expression

e

Specifies that the input is a single classic expression. This option is mutually exclusive with the `-f` option, which specifies that the input is a NetScaler configuration file.

classic expression

The classic expression that you want to convert to the default syntax.

Converting a NetScaler Configuration File

You can use the `nspepi` tool to convert all the classic expressions in a NetScaler configuration file to the default syntax (except for those commands that do not support the default syntax). The `nspepi` tool must be run from the shell prompt on the NetScaler appliance.

To convert all the classic expressions in a NetScaler configuration file to the default syntax by using the NetScaler command line interface

At the shell prompt, type:

```
nspepi -f "<ns config file>" -v
```

Example

```
root@NS# nspepi -f ns.conf
OUTPUT: New configuration file created: new_ns.conf
OUTPUT: New warning file created: warn_ns.conf
WARNINGS: Total number of warnings due to bind commands: 18
WARNINGS: Line numbers which has bind command issues: 305, 306, 706, 707, 708, 709, 710, 711, 712, 713,
714, 715, 767, 768, 774, 775, 776, 777
root@NS#
```

Parameters for converting the classic expressions in a NetScaler configuration file to the default syntax

f

Specifies that the input is a NetScaler configuration file. This option is mutually exclusive with the `-e` option, which specifies that the input is a single classic expression.

ns config file

The full path to the NetScaler configuration file. If the NetScaler configuration file is in the present working directory, the name of the NetScaler configuration file is sufficient.

v

The verbose option. If this option is specified, the output of the conversion tool is printed to the screen. The configuration file and the warning file are created even if this option is used.

Conversion Warnings

When classic expressions that are included in CLI commands are upgraded to the default syntax, the number of characters in the expression might exceed the 1499-character limit. The commands that include expressions longer than 1499 characters fail when the configuration is being applied. You must manually update these commands.

In addition, multiple classic policies can be bound to a given bind point with priority 0 or with equal priority, but the default syntax policy infrastructure does not support a priority value of 0 or policies with the same priority at a given bind point. These commands fail when the configuration is being applied. The commands must be updated manually with the correct priority values.

The line numbers of lines that threw a warning during conversion are listed at the end of the output in a warning line. In addition, a warning file is created in the same directory as the one in which the old and new configuration files reside. The name of the warning file is the same as the name of the input configuration file except that the string `warn_` is added as a prefix.

About Migration from Classic to Default Syntax Policies and Expressions

The NetScaler supports either classic or default syntax policies within a feature. You cannot have both types in the same feature. Over the past few releases, some NetScaler features have migrated from using classic policies and expressions to default syntax policies and expressions. If a feature of interest to you has changed to the default syntax format, you may have to manually migrate the older information. Following are guidelines for deciding if you need to migrate your policies:

- If you configured classic policies in a version of the Integrated Caching feature prior to release 9.0 and then upgrade to version 9.0 or later, there is no impact. All legacy policies are migrated to the default syntax policy format.
- For other features, you need to manually migrate classic policies and expressions to the default syntax if the feature has migrated to the default syntax.

Before You Proceed

Before configuring expressions and policies, be sure you understand the relevant NetScaler feature and the structure of your data, as follows:

- Read the documentation on the relevant feature.
- Look at the data stream for the type of data that you want to configure.

You may want to run a trace on the type of traffic or content that you want to configure. This will give you an idea of the parameters and values, and operations on these parameters and values, that you need to specify in an expression.

Configuring Default Syntax Policies

You can create default syntax policies for various NetScaler features, including DNS, Rewrite, Responder, and Integrated Caching, and the clientless access function in the Access Gateway. Policies control the behavior of these features.

When you create a policy, you assign it a name, a rule (an expression), feature-specific attributes, and an action that is taken when data matches the policy. After creating the policy, you determine when it is invoked by binding it globally or to either request-time or response-time processing for a virtual server.

Policies that share the same bind point are known as a *policy bank*. For example, all policies that are bound to a virtual server constitute the policy bank for the virtual server. When binding the policy, you assign it a priority level to specify when it is invoked relative to other policies in the bank. In addition to assigning a priority level, you can configure an arbitrary evaluation order for policies in a bank by specifying Goto expressions.

In addition to policy banks that are associated with a built-in bind point or a virtual server, you can configure *policy labels*. A policy label is a policy bank that is identified by an arbitrary name. You invoke a policy label, and the policies in it, from a global or virtual-server-specific policy bank. A policy label or a virtual-server policy bank can be invoked from multiple policy banks.

For some features, you can use the policy manager to configure and bind policies.

Rules for Names in Identifiers Used in Policies

The names of identifiers in the named expression, HTTP callout, pattern set, and rate limiting features must begin with an ASCII alphabet or an underscore (_). The remaining characters can be ASCII alphanumeric characters or underscores (_).

The names of these identifiers must not begin with the following reserved words:

The words ALT, TRUE, or FALSE or the Q or S one-character identifier.

- The special-syntax indicator RE (for regular expressions) or XP (for XPath expressions).
- Expression prefixes, which currently are the following:
  - CLIENT
  - EXTEND
  - HTTP
  - SERVER
  - SYS
  - TARGET
  - TEXT
  - URL
  - MYSQL
  - MSSQL

Additionally, the names of these identifiers cannot be the same as the names of enumeration constants used in the policy infrastructure. For example, the name of an identifier cannot be IGNORECASE, YEAR, or LATIN2_CZECH_CS (a MySQL character set).

Note: The NetScaler appliance performs a case-insensitive comparison of identifiers with these words and enumeration constants. For example, names of the identifiers cannot begin with TRUE, True, or true.

Creating or Modifying a Policy

All policies have some common elements. Creating a policy consists, at minimum, of naming the policy and configuring a rule. The policy configuration tools for the various features have areas of overlap, but also differences. For the details of configuring a policy for a particular feature, including associating an action with the policy, see the documentation for the feature.

To create a policy, begin by determining the purpose of the policy. For example, you may want to define a policy that identifies HTTP requests for image files, or client requests that contain an SSL certificate. In addition to knowing the type of information that you want the policy to work with, you need to know the format of the data that the policy is analyzing.

Next, determine whether the policy is globally applicable, or if it pertains to a particular virtual server. Also consider the effect that the order in which your policies are evaluated (which will be determined by how you bind the policies) will have on the policy that you are about to configure.

To create a policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a policy and verify the configuration:

- `add responder|dns|cs|rewrite|cache` **`policy`** `<policyName>` **`-rule`** `<expression> [<feature-specific information>]`
- `show rewrite` **`policy`** `pol_remove-ae`

Example 1:

```
add rewrite policy "pol_remove-ae" true "act_remove-ae"
 Done
> show rewrite policy pol_remove-ae
      Name: pol_remove-ae
      Rule: true
      RewriteAction: act_remove-ae
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
      Bound to: GLOBAL RES_OVERRIDE
      Priority: 90
      GotoPriorityExpression: END
 Done
>
```

Example 2:

```
add cache policy BranchReportsCachePolicy -rule q{http.req.url.query.value("actionoverride").contains("
Done
  show cache policy BranchReportsCachePolicy
     Name: BranchReportsCachePolicy
     Rule: http.req.url.query.value("actionoverride").contains("branchReports")
     CacheAction: CACHE
     Stored in group: DEFAULT
     UndefAction: Use Global
     Hits: 0
     Undef Hits: 0
Done
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, see Configuring Default Syntax Expressions in a Policy.

Parameters for creating or modifying a policy

policyName

A unique name for the policy. (Cannot be changed for an existing policy.)

Note that in the Content Switching feature, the name cannot start with app_ because this is a reserved name. Policies with this name are not displayed in the configuration utility.

expression

A logical expression. See Configuring Default Syntax Expressions: Getting Started.

feature-specific information

Varies by feature. Includes a built-in or user-defined action that you associate with the policy. See the documentation for the feature to which the policy applies.

To create or modify a policy by using the configuration utility

1. In the navigation pane, expand the name of the feature for which you want to configure a policy, and then click **Policies**. For example, you can select **Content Switching**, **Integrated Caching**, **DNS**, **Rewrite**, or **Responder**.

2. In the details pane, click **Add**, or select an existing policy and click **Open**. A **policy configuration** dialog box appears.

3. Specify values for the following parameters. (An asterisk indicates a required parameter. For a term in parentheses, see the corresponding parameter in "Parameters for creating or modifying a policy.")

4. Click **Create**, and then click **Close**.

5. Click **Save**. A policy is added.

   Note: After you create a policy, you can view the policy's details by clicking the policy entry in the configuration pane. Details that are highlighted and underlined are links to the corresponding entity (for example, a named expression).

Policy Configuration Examples

These examples show how policies and their associated actions are entered at the NetScaler command line. In the configuration utility, the expressions would appear in the Expression window of the feature-configuration dialog box for the integrated caching or rewrite feature.

Following is an example of creating a caching policy. Note that actions for caching policies are built in, so you do not need to configure them separately from the policy.

```
add cache policy BranchReportsCachePolicy -rule q{http.req.url.query.value("actionoverride").contains("bra
```

Following is an example of a Rewrite policy and action:

```
add rewrite action myAction1 INSERT_HTTP_HEADER "myHeader" "valueForMyHeader"
add rewrite policy myPolicy1 "http.req.url.contains(\"myURLstring\")" myAction1
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, see Configuring Default Syntax Expressions in a Policy.

Binding Policies That Use the Default Syntax

After defining a policy, you indicate when the policy is to be invoked by binding the policy to a bind point and specifying a priority level. You can bind a policy to only one bind point. A bind point can be global, that is, it can apply to all virtual servers that you have configured. Or, a bind point can be specific to a particular virtual server, which can be either a load balancing or a content switching virtual server. Not all bind points are available for all features.

The order in which policies are evaluated determines the order in which they are applied, and features typically evaluate the various policy banks in a particular order. Sometimes, however, other features can affect the order of evaluation. Within a policy bank, the order of evaluation depends on the values of parameters configured in the policies. Most features apply all of the actions associated with policies whose evaluation results in a match with the data that is being processed. The integrated caching feature is an exception.

Feature-Specific Differences in Policy Bindings

You can bind policies to built-in, global bind points (or banks), to virtual servers, or to policy labels.

However, the NetScaler features differ in terms of the types of bindings that are available. The following table summarizes how you use policy bindings in various NetScaler features that use policies.

Table 1. Feature-Specific Bindings for Policies

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Po the Featu
DNS	none	DNS policies	Global	To deter how to p DNS resol requests.

Content Switching **Note:** This feature can support either or classic policies or policies that use the default syntax, but not both.	Content Switching (CS)	Content Switching policies	• Content switching or cache redirection virtual server • Policy label	To deter what serv group of is respon serving response on charac of an inc request. Request character include d type, lan cookies, method, type, and associate server.
Integrated Caching	none	Caching policies	• Global override • Global default • Policy label • Load balancing, content switching, or SSL offload virtual server	To deter whether response stored in served fr NetScale integrate
Responder	none	Responder policies	• Global override • Global default • Policy label • Load balancing, content switching, or SSL offload virtual server	To config behavior Respond function.

Rewrite	none	Rewrite policies	• Global override • Global default • Policy label • Load balancing, content switching, or SSL offload virtual server	To identi data that want to r before se The polic provide r modifying data. For exam can modi data to re request t selected based on address o incoming or to mas informati response security purposes.
URL Transform function in the Rewrite feature	none	Transformation policies	• Global override • Global default • Policy label	To identi in HTTP transacti text files purpose o evaluatin whether should be altered.
Access Gateway (clientless VPN functions only)	VPN server	Clientless Access policies	• VPN Global • VPN server	To deter how the A Gateway performs authentic authoriza auditing, other fun and to de rewrite r general W access us Access Ga

Bind Points and Order of Evaluation

For a policy to take effect, you must ensure that the policy is invoked at some point during processing. To do so, you associate the policy with a bind point. The collection of policies that is bound to a bind point is known as a policy bank.

Following are the bind points that the NetScaler evaluates, listed in the typical order of evaluation within a policy bank

1. **Request-time override.** When a request flows through a feature, the NetScaler first evaluates request-time override policies for the feature.

2. **Request-time Load Balancing virtual server.** If policy evaluation cannot be completed after all the request-time override policies have been evaluated, the NetScaler processes request-time policies for load balancing virtual servers.

3. **Request-time Content Switching virtual server.** If policy evaluation cannot be completed after all the request-time policies for load balancing virtual servers have been evaluated, the NetScaler processes request-time policies for content switching virtual servers.

4. **Request-time default.** If policy evaluation cannot be completed after all request-time, virtual server-specific policies have been evaluated, the NetScaler processes request-time default policies.

5. **Response-time override.** At response time, the NetScaler starts with policies that are bound to the response-time override bind point.

6. **Response-time Load Balancing virtual server.** If policy evaluation cannot be completed after all response-time override policies have been evaluated, the NetScaler process the response-time policies for load balancing virtual servers.

7. **Response-time Content Switching virtual server.** If policy evaluation cannot be completed after all policies have been evaluated for load balancing virtual servers, the NetScaler process the response-time policies for content switching virtual servers.

8. **Response-time default.** If policy evaluation cannot be completed after all response-time, virtual-server-specific policies have been evaluated, the NetScaler processes response-time default policies.

Policy Evaluation across Features

In addition to attending to evaluation of policies within a feature, if you have bound policies to a content switching virtual server, note that these policies are evaluated before other policies. Binding a policy to a content switching vserver produces a different result in NetScaler versions 9.0.x and later than in 8.x versions. In NetScaler 9.0 and later versions, evaluation occurs as follows:

- Content switching policies are evaluated before other policies. If a content switching policy evaluates to TRUE, the target load balancing vserver is selected.

- If all content switching policies evaluate to FALSE, the default load balancing vserver under the content switching VIP is selected.

After a target load balancing vserver is selected by the content switching process, policies are evaluated in the following order:

1. Policies that are bound to the global override bind point.

2. Policies that are bound to the default load balancing vserver.

3. Policies that are bound to the target content switching vserver.

4. Policies that are bound to the global default bind point.

To be sure that the policies are evaluated in the intended order, follow these guidelines:

- Make sure that the default load balancing vserver is not directly reachable from the outside; for example, the vserver IP address can be 0.0.0.0.
- To prevent exposing internal data on the load balancing default vserver, configure a policy to respond with a “503 Service Unavailable” status and bind it to the default load balancing vserver.

Entries in a Policy Bank

Each entry in a policy bank has, at minimum, a policy and a priority level. You can also configure entries that change the priority-based evaluation order, and you can configure entries that invoke external policy banks.

The following table summarizes each entry in a policy bank.

Table 2. Format of Each Entry in a Policy Bank

Policy Name	Priority	Goto Expression	Invocation Type	Policy Bank to Be Invoked
The policy name, or a “dummy” policy named NOPOLICY. The NOPOLICY entry controls evaluation flow without processing a rule.	An integer.	Optional. Identifies the next policy in the bank to evaluate, or ends any further evaluation	Optional. Indicates that an external policy bank will be invoked. This field restricts the choices to a global policy label or a virtual server.	Optional. Used with Invocation Type. This is the label for a policy bank or a virtual server name. The NetScaler returns to the current bank after processing the external bank.

If the policy evaluates to TRUE, the NetScaler stores the action that is associated with the policy. If the policy evaluates to FALSE, the NetScaler evaluates the next policy. If the policy is neither TRUE nor FALSE, the NetScaler uses the associated Undef (undefined) action.

Evaluation Order within a Policy Bank

Within a policy bank, the evaluation order depends on the following items:

A priority.

The most minimal amount of information about evaluation order is a numeric priority level. The lower the number, the higher the priority.

A Goto expression.

If supplied, the Goto expression indicates the next policy to be evaluated, typically within the same policy bank.. Goto expressions can only proceed forward in a bank. To prevent looping, a policy bank configuration is not valid if a Goto statement points backwards in the bank.

Invocation of other policy banks.

Any entry can invoke an external policy bank. The NetScaler provides a built-in entity named NOPOLICY that does not have a rule. You can add a NOPOLICY entry in a policy bank when you want to invoke another policy bank, but do not want to process any other rules prior to the invocation. You can have multiple NOPOLICY entries in multiple policy banks.

Values for a Goto expression are as follows:

NEXT.

This keyword selects the policy with the next higher priority level in the current policy bank.

An integer.

If you supply an integer, it must match the priority level of another policy in the current policy bank.

END.

This keyword stops evaluation after processing the current policy, and no additional policies in this bank are processed.

Blank.

If the Goto expression is empty, it is the same as specifying END.

A numeric expression.

This is a default syntax expression that resolves to a priority number for another policy in the current bank.

USE_INVOCATION_RESULT.

This phrase can be used only if you are invoking an external policy bank. Entering this phrase causes the NetScaler to perform one of the following actions:

- If the final Goto in the invoked policy bank has a value of END or is empty, the invocation result is END, and evaluation stops.
- If the final Goto expression in the invoked policy bank is anything other than END, the NetScaler performs a NEXT.

The following table illustrates a policy bank that uses Goto statements and policy bank invocations.

Table 3. Example of a Policy Bank That Uses Gotos and External Bank Invocations

Policy Name	Priority	Goto	Invocation	Policy Bank to Be Invoked
ClientCertificatePolicy (rule: does the request contain a client certificate?)	100	300	None	None
SubnetPolicy (rule: is the client from a private subnet?)	200	NEXT	None	None
NOPOLICY	300	USE INVOCATION RESULT	Request vserver	My_Request_VServer
NOPOLICY	350	USE INVOCATION RESULT	Policy Label	My_Policy_Label
WorkingHoursPolicy (rule: is it working hours?)	400	END	None	None

How Policy Evaluation Ends

Evaluation of a policy bank ends when one of the following takes place:

- A policy evaluates to TRUE and its Goto statement value is END.

  No further policies or policy banks in this feature are evaluated.

- An external policy bank is invoked, its evaluation returns an END, and the Goto statement uses a value of USE_INVOCATION_RESULT or END.

  Evaluation continues with the next policy bank for this feature. For example, if the current bank is the request-time override bank, the NetScaler next evaluates request-time policy banks for the virtual servers.

- The NetScaler has walked through all the policy banks in this feature, but has not encountered an END.

If this is the last entry to be evaluated in this policy bank, the NetScaler proceeds to the next feature.

How Features Use Actions after Policy Evaluation

After evaluating all relevant policies for a particular data point (for example, an HTTP request), the NetScaler stores all the actions that are associated with any policy that matched the data.

For most features, all the actions from matching policies are applied to a traffic packet as it leaves the NetScaler. The Integrated Caching feature only applies one action: CACHE or NOCACHE. This action is associated with the policy with the lowest priority value in the "highest priority" policy bank (for example, request-time override policies are applied before virtual server-specific policies).

Binding a Policy Globally

The following binding procedures are typical. However, refer to the documentation for the feature of interest to you for complete instructions.

To bind an Integrated Caching policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind an Integrated Chaching policy and verify the configuration:

- `bind cache global <policyName> -priority <positiveInteger> [-type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]`
- `show cache global`

  Example

  ```
  bind cache global _nonPostReq -priority 100 -type req_default
  Done
  > show cache global
  1)      Global bindpoint: REQ_DEFAULT
         Number of bound policies: 2

  2)      Global bindpoint: RES_DEFAULT
         Number of bound policies: 1
  Done
  ```

  The type argument is optional to maintain backward compatibility. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression.

To bind a Rewrite policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a Rewrite policy and verify the configuration:

- `bind rewrite global <policyName> <positiveIntegerAsPriority> [-type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]`
- `show rewrite global`

  Example

```
bind rewrite global pol_remove-pdf 100
 Done
> show rewrite global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

 Done
```

The type argument is optional for globally bound policies, to maintain backward compatibility. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression.

To bind a compression policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a compression policy and verify the configuration:

- `bind cmp global <policyName> -priority <positiveInteger> [-type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]`
- `show cmp global`

Example

```
> bind cmp global cmp_pol_1 -priority 100
 Done
> show cmp policy cmp_pol_1
        Name: cmp_pol_1
        Rule: HTTP.REQ.URL.SUFFIX.EQ("BMP")
        Response Action: COMPRESS
        Hits: 0

        Policy is bound to following entities
        1) GLOBAL REQ_DEFAULT
        Priority: 100
        GotoPriorityExpression: END
 Done
>
```

To bind a Responder policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a Responder policy and verify the configuration:

- `bind responder global <policyName> <positiveIntegerAsPriority> [-type OVERRIDE | DEFAULT ]`
- `show responder global`

  Example

```
 bind responder global pol404Error1 200
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
```

To bind a DNS policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a DNS policy and verify the configuration:

- `bind dns global <policyName> <positiveIntegerAsPriority>`
- `show dns global`

  Example

```
> bind dns global pol_ddos_drop1 150
 Done
> show dns global
Policy name : pol_ddos_drop
        Priority : 100
        Goto expression : END
Policy name : pol_ddos_drop1
        Priority : 150
 Done
>
```

To bind an Integrated Caching, Responder, Rewrite, or Compression policy globally by using the configuration utility

1. In the navigation pane, click the name of the feature for which you want to bind the policy.
2. In the details pane, click **<Feature Name> policy manager.**
3. In the **Policy Manager** dialog box, select the bind point to which you want to bind the policy (for example, for Integrated Caching, Rewrite, or Compression, you could select **Request** and **Default Global**). The Responder does not differentiate between request-time and response-time policies.
4. Click **Insert Policy** and, from the **Policy Name** pop-up menu, select the policy name. A priority is assigned automatically to the policy, but you can click the cell in the **Priority** column and drag it anywhere within the dialog box if you want the policy to be evaluated after other policies in this bank. The priority is automatically reset. Note that priority values within a policy bank must be unique.
5. Click **Apply Changes**.
6. Click **Close**. A message in the status bar indicates that the policy is binded successfully.

To bind a DNS policy globally by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **global bindings** dialog box, click **Insert Policy**, and select the policy that you want to bind globally.
4. Click in the **Priority** field and enter the priority level.
5. Click **OK**. A message in the status bar indicates that the policy is binded successfully.

Binding a Policy to a Virtual Server

A globally bound policy applies to all load balancing and content switching virtual servers.

Note that when binding a policy to a virtual server, you must identify it as a request-time or a response-time policy.

To bind a policy to a load balancing or content switching virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a policy to a load balancing or content switching virtual server and verify the configuration:

- `bind lb|cs vserver <virtualServerName> -policyName <policyName> -priority <positiveInteger> -type REQUEST|RESPONSE`
- `show lb vserver lbvip`

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
        Time since last state change: 28 days, 01:57:26.350
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:50
2)      Policy : cf-pol Priority:1      Inherited
 Done
```

To bind a policy to an SSL offload virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to bind a policy to an SSL offload virtual server and verify the configuration:

```
bind ssl vserver <virtualServerName> -policyName <policyName>
-priority <positiveInteger>
```

To bind a policy to a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, **Content Switching**, **SSL Offload**, **AAA-Application Traffic**, or **Access Gateway**, and then click **Virtual Servers**.
2. In the details pane, double-click the virtual server to which you want to bind the policy, and then click **Open**.
3. On the **Policies** tab, click the icon for the type of policy that you want to bind (the choices are feature-specific), and then click the name of the policy. Note that for some features, you can bind both classic policies and policies that use the default syntax to the virtual server.
4. If you are binding a policy to a Content Switching virtual server, in the **Target** field select a load balancing virtual server to which traffic that matches the policy is sent.
5. Click **OK**. A message in the status bar indicates that the policy is binded successfully.

Displaying Policy Bindings

You can display policy bindings to verify that they are correct.

To display policy bindings by using the NetScaler command line

At the NetScaler command prompt, type the following commands to display policy bindings and verify the configuration:

```
show <featureName> policy <policyName>
```

Example

```
> show rewrite policy pol_remove-pdf
      Name: pol_remove-pdf
      Rule: http.req.url.contains(".pdf")
      RewriteAction: act_remove-ae
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
      Bound to: GLOBAL REQ_DEFAULT
      Priority: 100
      GotoPriorityExpression: END
 Done
>
```

To display global policy bindings for Integrated Caching, Rewrite, or Responder by using the configuration utility

1. In the navigation pane, expand the feature that contains the policy that you want to view, and then click **Policies**.
2. In the details pane, click the policy. Bound policies have a check mark next to them.
3. At the bottom of the page, under **Details**, next to **Bound to**, view the entity to which the policy is bound.

To display global policy bindings for DNS or Clientless Access in the Access Gateway by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Policies**.
2. In the details pane, click **Global Bindings**.

To display global policy bindings for Content Switching by using the configuration utility

1. In the navigation pane, expand **Content Switching**, and then click **Policies**.
2. In detailed pane, select policy.
3. In the details pane, click **Show Bindings**.

Unbinding a Policy

If you want to re-assign a policy or delete it, you must first remove its binding.

To unbind an integrated caching, rewrite, or compression default syntax policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind an integrated caching, rewrite, or compression default syntax policy globally and verify the configuration:

- `unbind cache|rewrite|cmp global <policyName> [-type req_override|req_default|res_override|res_default] [-priority <positiveInteger>]`
- `show cache|rewrite|cmp global`

Example

```
> unbind cache global _nonPostReq
 Done
> show cache global
1)      Global bindpoint: REQ_DEFAULT
       Number of bound policies: 1

2)      Global bindpoint: RES_DEFAULT
       Number of bound policies: 1

 Done
```

The priority is required only for the “dummy” policy named NOPOLICY.

To unbind a responder policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a responder policy globally and verify the configuration:

- `unbind responder global <policyName> [-type override|default] [-priority <positiveInteger>]`
- `show responder global`

Example

```
> unbind responder global pol404Error
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
```

The priority is required only for the “dummy” policy named NOPOLICY.

To unbind a DNS policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a DNS policy globally and verify the configuration:

- `unbind responder global <policyName>`
- `unbind responder global`

Example

```
 unbind dns global dfgdfg
 Done
 show dns global
Policy name : dfgdfggfhg
        Priority : 100
        Goto expression : END
 Done
```

To unbind a default syntax policy from a virtual server by using the NetScaler command line

At the NetScaler command prompt, type the following commands to unbind a default syntax policy from a virtual server and verify the configuration:

- `unbind <featureType> vserver <virtualServerName> -policyName <policyName> [-priority <positiveInteger>] [-type REQUEST|RESPONSE]`
- `show lb vserver lbvip`

Example

```
 unbind cs vserver vs-cont-switch  -policyName pol1
 Done
> show cs vserver vs-cont-switch
        vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
        State: UP
        Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
```

```
        Time since last state change: 0 days, 02:47:55.750
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Vserver IP and Port insertion: OFF
        Case Sensitivity: ON
        Push: DISABLED  Push VServer:
        Push Label Rule: none
Done
```

The priority is required only for the “dummy” policy named NOPOLICY.

To unbind an integrated caching, responder, rewrite, or compression default syntax policy globally by using the configuration utility

1. In the navigation pane, click the feature with the policy that you want to unbind (for example, **Integrated Caching**).
2. In the details pane, click **<Feature Name> policy manager**.
3. In the **Policy Manager** dialog box, select the bind point with the policy that you want to unbind, for example, **Default Global**.
4. Click the policy name that you want to unbind, and then click **Unbind Policy**.
5. Click **Apply Changes**.
6. Click **Close**. A message in the status bar indicates that the policy is unbound successfully.

To unbind a DNS policy globally by using the configuration utility

1. In the navigation pane, expand **DNS**, and then click **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **Global Bindings** dialog box, select policy and clilck **unbind policy**.
4. Click **OK**. A message in the status bar indicates that the policy is unbinded successfully.

To unbind a default syntax policy from a load balancing or content switching virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing** or **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, double-click the virtual server from which you want to unbind the policy.
3. On the **Policies** tab, in the **Active** column, clear the check box next to the policy that you want to unbind.
4. Click **OK**. A message in the status bar indicates that the policy is unbinded successfully.

Creating Policy Labels

In addition to the built-in bind points where you set up policy banks, you can also configure user-defined policy labels and associate policies with them.

Within a policy label, you bind policies and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. The NetScaler also permits you to define an arbitrary evaluation order as follows:

- You can use “goto” expressions to point to the next entry in the bank to be evaluated after the current one.
- You can use an entry in a policy bank to invoke another bank.

Creating Policy Labels

Each feature determines the type of policy that you can bind to a policy label, the type of load balancing virtual server that you can bind the label to, and the type of content switching virtual server from which the label can be invoked. For example, a TCP policy label can only be bound to a TCP load balancing virtual server. You cannot bind HTTP policies to a policy label of this type. And you can invoke a TCP policy label only from a TCP content switching virtual server.

After configuring a new policy label, you can invoke it from one or more banks for the built-in bind points.

To create a caching policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a Caching policy label and verify the configuration:

- `add cache policylabel <policyLabelName> -evaluates req|res`
- `show cache policylabel lbl-cache-pol`

Example

```
> add cache policylabel lbl-cache-pol -evaluates req
 Done

> show cache policylabel lbl-cache-pol
      Label Name: lbl-cache-pol
      Evaluates: REQ
      Number of bound policies: 0
      Number of times invoked: 0
 Done
>
```

To create a Content Switching policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a Content Switching policy label and verify the configuration:

- `add cs policylabel <policyLabelName> http|tcp|rtsp|ssl`
- `show cs policylabel <policyLabelName>`

Example

```
> add cs policylabel lbl-cs-pol http
 Done
> show cs policylabel lbl-cs-pol
        Label Name: lbl-cs-pol
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
 Done
```

To create a Rewrite policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a Rewrite policy label and verify the configuration:

- `add rewrite policylabel <policyLabelName> http_req|http_res|url|text|clientless_vpn_req|clientless_vpn_res`
- `show rewrite policylabel <policyLabelName>`

Example

```
> add rewrite policylabel lbl-rewrt-pol http_req
 Done

> show rewrite policylabel lbl-rewrt-pol
        Label Name: lbl-rewrt-pol
        Transform Name: http_req
        Number of bound policies: 0
        Number of times invoked: 0
 Done
```

To create a Responder policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to create a Responder policy label and verify the configuration:

- `add responder policylabel <policyLabelName>`
- `show responder policylabel <policyLabelName>`

Example

```
> add responder policylabel lbl-respndr-pol
 Done
```

```
> show responder policylabel lbl-respndr-pol
        Label Name: lbl-respndr-pol
        Number of bound policies: 0
        Number of times invoked: 0
 Done
```

Note: Invoke this policy label from a policy bank. For more information, see Binding a Policy to a Policy Label.

To create a policy label by using the configuration utility

1. In the navigation pane, expand the feature for which you want to create a policy label, and then click **Policy Labels**. The choices are **Integrated Caching**, **Rewrite**, **Content Switching**, or **Responder**.

2. In the details pane, click **Add**.

3. In the **Name** box, enter a unique name for this policy label.

4. Enter feature-specific information for the policy label. For example, for Integrated Caching, in the **Evaluates** drop-down menu, you would select **REQ** if you want this policy label to contain request-time policies, or select **RES** if you want this policy label to contain response-time policies. For Rewrite, you would select a **Transform** name. For information on integrated caching, see the *Citrix NetScaler Application Optimization Guide* at http://support.citrix.com/article/CTX128681. For information on rewrite, see the *Citrix NetScaler Application Security Guide* at http://support.citrix.com/article/CTX128674.

5. Click **Create**.

6. Configure one of the built-in policy banks to invoke this policy label. For more information, see Binding a Policy to a Policy Label. A message in the status bar indicates that the policy label is created successfully.

Binding a Policy to a Policy Label

As with policy banks that are bound to the built-in bind points, each entry in a policy label is a policy that is bound to the policy label. As with policies that are bound globally or to a vserver, each policy that is bound to the policy label can also invoke a policy bank or a policy label that is evaluated after the current entry has been processed. The following table summarizes the entries in a policy label.

Name

The name of a policy, or, to invoke another policy bank without evaluating a policy, the “dummy” policy name NOPOLICY.

You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.

Priority

An integer. This setting can work with the Goto expression.

Goto Expression

Determines the next policy to evaluate in this bank. You can provide one of the following values:

NEXT:

Go to the policy with the next higher priority.

END:

Stop evaluation.

USE_INVOCATION_RESULT:

Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT.

Positive number:

The priority number of the next policy to be evaluated.

Numeric expression:

An expression that produces the priority number of the next policy to be evaluated.

The Goto can only proceed forward in a policy bank.

If you omit the Goto expression, it is the same as specifying END.

Invocation Type

Designates a policy bank type. The value can be one of the following:

Request Vserver:

Invokes request-time policies that are associated with a virtual server.

Response Vserver:

Invokes response-time policies that are associated with a virtual server.

Policy label:

Invokes another policy bank, as identified by the policy label for the bank.

Invocation Name

The name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

Configuring a Policy Label or Virtual Server Policy Bank

After you have created policies, and created policy banks by binding the policies, you can perform additional configuration of polices within a label or policy bank. For example, before you configure invocation of an external policy bank, you might want to wait until you have configured that policy bank.

Configuring a Policy Label

A policy label consists of a set of policies and invocations of other policy labels and virtual server-specific policy banks. An Invoke parameter enables you to invoke a policy label or a virtual server-specific policy bank from any other policy bank. A special-purpose NoPolicy entry enables you to invoke an external bank without processing an expression (a rule). The NoPolicy entry is a “dummy” policy that does not contain a rule.

For configuring policy labels from the NetScaler command line, note the following elaborations of the command syntax:

- gotoPriorityExpression is configured as described in Entries in a Policy Bank.
- The type argument is required. This is unlike binding a conventional policy, where this argument is optional.
- You can invoke the bank of policies that are bound to a virtual server by using the same method as you use for invoking a policy label.

To configure a policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a pliccy label and verify the configuration:

- `bind cache|rewrite|responder policylabel <policylabelName> -policyName <policyName> -priority <priority> [-gotoPriorityExpression <gotopriorityExpression>] [-invoke reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>]`
- `show cache|rewrite|responder policylabel <policylabelName>`

Example

```
bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -priority 100
Done
show cache policylabel _reqBuiltinDefaults
      Label Name: _reqBuiltinDefaults
      Evaluates: REQ
      Number of bound policies: 3
      Number of times invoked: 0
1)    Policy Name: _nonGetReq
      Priority: 100
      GotoPriorityExpression: END
2)    Policy Name: _advancedConditionalReq
      Priority: 200
```

```
        GotoPriorityExpression: END

3)      Policy Name: _personalizedReq
        Priority: 300
        GotoPriorityExpression: END
 Done
```

To invoke a policy label from a Rewrite policy bank with a NOPOLICY entry by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a policy label from a Rewrite policy bank with a NOPOLICY entry and verify the configuration:

- `bind rewrite global NOPOLICY <priority> -gotoPriorityExpression <gotopriorityExpression> -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>`

- `show rewrite global`

  Example

```
> bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke policylabel lbl-rewrt-pol
 Done
> show rewrite global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1
 Done
```

To invoke a policy label from an Integrated Caching policy bank by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a policy label from an Integrated Caching policy bank and verify the configuration:

- `bind cache global NOPOLICY -priority <priority> -gotoPriorityExpression <gotopriorityExpression> -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>`

- `show cache global`

  Example

```
bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -type REQ_DEFAULT -invoke policy
Done
> show cache global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 2

2)      Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

Done
```

To invoke a policy label from a Responder policy bank by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a policy label from a Responder policy bank and verify the configuration:

- `bind responder global NOPOLICY <priority> <gotopriorityExpression> -type OVERRIDE|DEFAULT -invoke vserver|policylabel <policyLabelName>|<vserverName>`
- `show responder global`

Example

```
> bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke policylabel lbl-respndr-pol
Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 2

Done
```

To configure a policy label by using the configuration utility

1. In the navigation pane, expand the feature for which you want to configure a policy label, and then click **Policy Labels**. The choices are **Integrated Caching**, **Rewrite**, or **Responder**.

2. In the details pane, double-click the label that you want to configure.

3. If you are adding a new policy to this policy label, click Insert Policy, and in the Policy Name field, select New Policy. For more information about adding a policy, see Creating or Modifying a Policy. Note that if you are invoking a policy bank, and do not want a rule to be evaluated prior to the invocation, click **Insert Policy**, and in the **Policy Name** field select **NOPOLICY**.

4. For each entry in this policy label, configure the following:

   Policy Name:

   This is already determined by the Policy Name, new policy, or NOPOLICY entry that you inserted in this bank.

   Priority:

   A numeric value that determines either an absolute order of evaluation within the bank, or is used in conjunction with a Goto expression.

   Expression:

   The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see Configuring Default Syntax Expressions: Getting Started.

   Action:

   The action to be taken if this policy evaluates to TRUE.

   Goto Expression:

   Optional. Used to augment the Priority level to determine the next policy or policy bank to evaluate. For more information on possible values for a Goto expression, see the table Entries in a Policy Bank.

   Invoke:

   Optional. Invokes another policy bank.

5. Click **Ok**. A message in the status bar indicates that the policy label is configured successfully.

Configuring a Policy Bank for a Virtual Server

You can configure a bank of policies for a virtual server. The policy bank can contain individual policies, and each entry in the policy bank can optionally invoke a policy label or a bank of policies that you configured for another virtual server. If you invoke a policy label or policy bank, you can do so without triggering an expression (a rule) by selecting a NOPOLICY “dummy” entry instead of a policy name.

To add policies to a virtual server policy bank by using the NetScaler command line

At the NetScaler command prompt, type the following commands to add policies to a virtual server policy bank and verify the configuration:

- `bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression <expression>] [-type REQUEST|RESPONSE]`
- `show lb|cs vserver <virtualServerName>`

Example

```
add lb vserver vs-cont-sw TCP
Done
show lb vserver vs-cont-sw
      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
      State: DOWN
      Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
      Time since last state change: 0 days, 00:02:14.420
      Effective State: DOWN
      Client Idle Timeout: 9000 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  0 (Total)      0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Connection Failover: DISABLED
Done
```

To invoke a policy label from a virtual server policy bank with a NOPOLICY entry by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a policy label from a virtual server policy bank with a NOPOLICY entry and verify the configuration:

- `bind lb|cs vserver <virtualServerName> -policyName NOPOLICY_REWRITE|NOPOLICY_CACHE|NOPOLICY_RESPONDER -priority <integer> -type REQUEST|RESPONSE -gotoPriorityExpression <gotoprioritiyExpression> -invoke reqVserver|resVserver|policyLabel <vserverName>|<labelName>`

- `show lb vserver`

  Example

  ```
  > bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200 -type REQUEST -gotoPriorityEx
   Done

  > show lb vserver
  1)      vserver-SSL-1 (10.102.29.50:443) - SSL  Type: ADDRESS
          State: DOWN[Certkey not bound]
          Last state change was at Tue Jul  7 10:50:53 2009 (+92 ms)
          Time since last state change: 36 days, 00:55:26.820
          Effective State: DOWN
          Client Idle Timeout: 180 sec
          Down state flush: ENABLED
          Disable Primary Vserver On Down : DISABLED
          No. of Bound Services :  0 (Total)       0 (Active)
          Configured Method: LEASTCONNECTION
          Mode: IP
          Persistence: NONE
          Vserver IP and Port insertion: OFF
          Push: DISABLED  Push VServer:
          Push Multi Clients: NO
          Push Label Rule: none
  2)      vsrvr-ssl-30 (10.102.29.45:443) - SSL   Type: ADDRESS
          State: DOWN[Certkey not bound]
          Last state change was at Wed Jul  8 07:52:20 2009 (+104 ms)
          Time since last state change: 35 days, 03:53:59.810
          Effective State: DOWN
          Client Idle Timeout: 180 sec
          Down state flush: ENABLED
          Disable Primary Vserver On Down : DISABLED
          No. of Bound Services :  0 (Total)       0 (Active)
          Configured Method: LEASTCONNECTION
          Mode: IP
          Persistence: NONE
          Vserver IP and Port insertion: OFF
          Push: DISABLED  Push VServer:
          Push Multi Clients: NO
  ```

```
    Push Label Rule: none
3)  ssl-vsrvr-30 (10.102.29.42:443) - SSL   Type: ADDRESS
    State: DOWN[Certkey not bound]
    Last state change was at Wed Jul  8 07:52:46 2009 (+856 ms)
    Time since last state change: 35 days, 03:53:33.60
    Effective State: DOWN
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  0 (Total)       0 (Active)
    Configured Method: LEASTCONNECTION
    Mode: IP
    Persistence: NONE
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none
4)  ssl-vsrvr-35 (10.102.29.35:443) - SSL   Type: ADDRESS
    State: DOWN[Certkey not bound]
    Last state change was at Wed Jul  8 07:59:48 2009 (+728 ms)
    Time since last state change: 35 days, 03:46:31.190
    Effective State: DOWN
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  1 (Total)       0 (Active)
    Configured Method: LEASTCONNECTION
    Mode: IP
    Persistence: NONE
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none
5)  Vserver-WSL (10.102.29.27:80) - HTTP    Type: ADDRESS
    State: DOWN
    Last state change was at Wed Jul 15 06:12:53 2009 (+9 ms)
    Time since last state change: 28 days, 05:33:26.910
    Effective State: DOWN
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Port Rewrite : DISABLED
    No. of Bound Services :  0 (Total)       0 (Active)
    Configured Method: LEASTCONNECTION
    Mode: IP
    Persistence: NONE
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none
6)  lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
    State: DOWN
    Last state change was at Wed Jul 15 05:54:24 2009 (+181 ms)
    Time since last state change: 28 days, 05:51:55.740
    Effective State: DOWN
    Client Idle Timeout: 180 sec
```

```
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
7)      vserver-LB-1 (0.0.0.0:0) - HTTP Type: ADDRESS
        State: DOWN
        Last state change was at Fri Aug  7 05:38:45 2009 (+413 ms)
        Time since last state change: 5 days, 06:07:34.510
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
 Done
```

To configure a virtual server policy bank by using the configuration utility

1. In the left navigation pane, expand **Load Balancing**, **Content Switching**, **SSL Offload**, **AAA - Application Traffic**, or **Access Gateway**, as appropriate, and then click **Virtual Servers**.

2. In the details pane, select the virtual server that you want to configure, and then click **Open**.

3. In the **Configure Virtual Server** dialog box click the **Policies** tab.

4. To create a new policy in this bank, click the icon for the type of policy or policy label that you want to add to the virtual server's bank of policies, click **Insert Policy**. Note that if you want to invoke a policy label without evaluating a policy rule, select the NOPOLICY "dummy" policy.

5. To configure an existing entry in this policy bank, enter the following:

   Priority:

   A numeric value that determines either an absolute order of evaluation within the bank or is used in conjunction with a Goto expression.

   Expression:

   The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see Configuring Default Syntax Expressions: Getting Started.

   Action: on:

   The action to be taken if this policy evaluates to TRUE.

   Goto Expression:

   Optional. Determines the next policy or policy bank to evaluate. For more information on possible values for a Goto expression, see Entries in a Policy Bank.

   Invoke:

   Optional. To invoke another policy bank, select the name of the policy label or virtual server policy bank that you want to invoke.

6. When you are done, click **OK**. A message in the status bar indicates that the policy is configured successfully.

Invoking or Removing a Policy Label or Virtual Server Policy Bank

Unlike a policy, which can only be bound once, you can use a policy label or a virtual server's policy bank any number of times by invoking it. Invocation can be performed from two places:

- From the binding for a named policy in a policy bank.
- From the binding for a NOPOLICY "dummy" entry in a policy bank.

Typically, the policy label must be of the same type as the policy from which it is invoked. For example, you would invoke a responder policy label from a responder policy.

Note: When binding or unbinding a global NOPOLICY entry in a policy bank at the command line, you specify a priority to distinguish one NOPOLICY entry from another.

To invoke a rewrite or integrated caching policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a rewrite or integrated caching policy label and verify the configuration:

- **bind cache|rewrite global** <policy_Name> **-priority** <positive_integer> [**-gotoPriorityExpression** <expression>] **-type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel** <label_name>
- **show cache|rewrite global**

  Example

  > bind cache global _nonPostReq2 -priority 100 -type req_override -invoke policylabel lbl-cache-pol
  Done
  > show cache global
  1) Global bindpoint: REQ_DEFAULT
  Number of bound policies: 2

  2) Global bindpoint: RES_DEFAULT
  Number of bound policies: 1

  3) Global bindpoint: REQ_OVERRIDE
  Number of bound policies: 1

  Done

To invoke a responder policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a responder policy label and verify the configuration:

- **bind responder global** <policy_Name> <priority_as_positive_integer> [<gotoPriorityExpression>] **-type REQ_OVERRIDE|REQ_DEFAULT|OVERRIDE|DEFAULT -invoke vserver|policylabel** <label_name>

- **show responder global**

  Example

  > bind responder global pol404Error1 300 -invoke policylabel lbl-respndr-pol
  Done
  > show responder global
  1) Global bindpoint: REQ_DEFAULT
  Number of bound policies: 2

  Done
  >

To invoke a Virtual Server Policy Bank by using the NetScaler command line

At the NetScaler command prompt, type the following commands to invoke a Virtual Server Policy Bank and verify the configuration:

- **bind lb vserver** <vserver_name> **-policyName** <policy_Name> **-priority** <positive_integer> [**-gotoPriorityExpression** <expression>] **-type REQUEST|RESPONSE -invoke reqvserver|resvserver|policylabel** <policy_Label_Name>

- **bind lb vserver** <vserver_name>

  Example

  > bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
  Done

  > show lb vserver lbvip
  lbvip (8.7.6.6:80) - HTTP Type: ADDRESS
  State: DOWN
  Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
  Time since last state change: 28 days, 06:37:49.250
  Effective State: DOWN
  Client Idle Timeout: 180 sec
  Down state flush: ENABLED
  Disable Primary Vserver On Down : DISABLED

```
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority: 100   Hits: 0

    1)      Policy : pol-ssl Priority:0
    2)      Policy : ns_cmp_msapp Priority:100
    3)      Policy : cf-pol Priority:1      Inherited
     Done
    >
```

To remove a rewrite or integrated caching policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to remove a rewrite or integrated caching policy label and verify the configuration:

- **unbind rewrite|cache global** NOPOLICY **-priority** <positiveInteger> **-type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT**
- **show rewrite|cache global**

  Example

```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
 Done
> show rewrite global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
```

To remove a responder policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to remove a responder policy label and verify the configuration:

- **unbind responder global** NOPOLICY **-priority** <positiveInteger> **-type OVERRIDE|DEFAULT**
- **show responder global**

Example

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
```

To remove a Virtual Server policy label by using the NetScaler command line

At the NetScaler command prompt, type the following commands to remove a Virtual Server policy label and verify the configuration:

- **unbind lb|cs vserver** <virtualServerName> **-policyName NOPOLICY-REWRITE|NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority** <positiveInteger>

- **show lb|cs vserver**

  Example

```
> unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
 Done
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
        Time since last state change: 28 days, 06:47:54.600
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority: 100   Hits: 0

1)      Policy : pol-ssl Priority:0
2)      Policy : cf-pol Priority:1      Inherited
 Done
```

To invoke a policy label or virtual server policy bank by using the configuration utility

1. Bind a policy, as described in Binding a Policy Globally, Binding a Policy to a Virtual Server, or Binding a Policy to a Policy Label. Alternatively, you can enter a NOPOLICY "dummy" entry instead of a policy name. You do this if you do not want to evaluate a policy before evaluating the policy bank.

2. In the **Invoke** field, select the name of the policy label or virtual server policy bank that you want to evaluate if traffic matches the bound policy. A message in the status bar indicates that the policy label or virtual server policy bank is invoked successfully.

To remove a policy label invocation by using the configuration utility

1. Open the policy and clear the Invoke field. Unbinding the policy also removes the invocation of the label. A message in the status bar indicates that the policy label is removed successfully.

Configuring and Binding Policies with the Policy Manager

Some applications provide a specialized Policy Manager in the NetScaler configuration utility to simplify configuring policy banks. It also lets you find and delete policies and actions that are not being used.

The Policy Manager is currently available for the Rewrite, Integrated Caching, Responder, and Compression features.

The following are keyboard equivalents for the procedures in this section:

- For editing a cell in the **Policy Manager**, you can tab to the cell and click F2 or press the SPACE bar on the keyboard.
- To select an entry in a drop-down menu, you can tab to the entry, press the space bar to view the drop-down menu, use the UP and DOWN ARROW keys to navigate to the entry that you want, and press the space bar again to select the entry.
- To cancel a selection in a drop-down menu, press the Escape key.
- To insert a policy, tab to the row above the insertion point and press Control + Insert, or click **Insert Policy**.
- To remove a policy, tab to the row that contains the policy and press Delete.

Note: Note that when you delete the policy, the NetScaler searches the Goto Expression values of other policies in the bank. If any of these Goto Expression values match the priority level of the deleted policy, they are removed.

To configure policy bindings by using the Policy Manager

1. In the navigation pane, click the feature for which you want to configure policies. The choices are **Responder**, **Integrated Caching**, **Rewrite** or **Compression**.

2. In the details pane, click **Policy Manager**.

3. If you are configuring classic policy bindings for compression, in the **Compression Policy Manager** dialog box, click **Switch to Classic Syntax**. The dialog box switches to the classic syntax view and displays the **Switch to Default Syntax** button. At any time before you complete configuring policy bindings, if you want to configure bindings for policies that use the default syntax, click the **Switch to Default Syntax** button.

4. For features other than **Responder**, to specify the bind point, click **Request** or **Response**, and then click one of the request-time or response-time bind points. The options are **Override Global**, **LB Virtual Server**, **CS Virtual Server**, **Default Global**, or **Policy Label**. If you are configuring the **Responder**, the **Request** and **Response** flow types are not available.

5. To bind a policy to this bind point, click **Insert Policy**, and select a previously configured policy, a **NOPOLICY** label, or the **New policy** option. Depending on the option that you select, you have the following choices:

   - **New policy:** Create the policy as described in Creating or Modifying a Policy, and then configure the priority level, GoTo expression, and policy invocation as described in the table, Format of Each Entry in a Policy Bank.

   - **Existing policy**, **NOPOLICY**, or **NOPOLICY<feature name>**: Configure the priority level, GoTo expression, and policy invocation as described in the table, Format of Each Entry in a Policy Bank. The **NOPOLICY** or **NOPOLICY<feature name>** options are available only for policies that use default syntax expressions.

6. Repeat the preceding steps to add entries to this policy bank.

7. To modify the priority level for an entry, you can do any of the following:

   - Double-click the **Priority** field for an entry and edit the value.

   - Click and drag a policy to another row in the table.

   - Click **Regenerate Priorities**.

   In all three cases, priority levels of all other policies are modified as needed to accommodate the new value. Goto Expressions with integer values are also updated automatically. For example, if you change a priority value of 10 to 100, all policies with a Goto Expression value of 10 are updated to the value 100.

8. To change the policy, action, or policy bank invocation for an row in the table, click the down arrow to the right of the entry and do one of the following:

   - To change the policy, select another policy name or select **New Policy** and follow the steps in Creating or Modifying a Policy.

   - To change the **Goto Expression**, select **Next**, **End**, **USE_INVOCATION_RESULT**, or select more and enter an expression whose result returns the priority level of

another entry in this policy bank.

- To modify an invocation, select an existing policy bank, or click New Policy Label and follow the steps in Binding a Policy to a Policy Label.

9. To unbind a policy or a policy label invocation from this bank, click any field in the row that contains the policy or policy label, and then click **Unbind Policy**.
10. When you are done, click **Apply Changes**. A message in the status bar indicates that the policy is bound successfully.

To remove unused policies by using the Policy Manager

1. In the navigation pane, click the feature for which you want to configure the policy bank. The choices are **Responder**, **Integrated Caching**, or **Rewrite**.
2. In the details pane, click **<Feature Name> policy manager**.
3. In the **<Feature Name> Policy Manage**r dialog box, click **Cleanup Configuration**.
4. In the **Cleanup Configuration** dialog box, select the items that you want to delete, and then click **Remove**.
5. In the **Remove** dialog box, click **Yes**.
6. Click **Close**. A message in the status bar indicates that the policy is removed successfully.

Configuring Default Syntax Expressions: Getting Started

Default syntax policies evaluate data on the basis of information that you supply in default syntax expressions. A default syntax expression analyzes data elements (for example, HTTP headers, source IP addresses, the NetScaler system time, and POST body data). In addition to configuring a default syntax expression in a policy, in some NetScaler features you configure default syntax expressions outside of the context of a policy.

To create a default syntax expression, you select a prefix that identifies a piece of data that you want to analyze, and then you specify an operation to perform on the data. For example, an operation can match a piece of data with a text string that you specify, or it can transform a text string into an HTTP header. Other operations match a returned string with a set of strings or a string pattern. You configure compound expressions by specifying Boolean and arithmetic operators, and by using parentheses to control the order of evaluation.

Default syntax expressions can also contain classic expressions. You can assign a name to a frequently used expression to avoid having to build the expression repeatedly.

Expression Characteristics

Policies and a few other entities include rules that the NetScaler uses to evaluate a packet in the traffic flowing through it, to extract data from the NetScaler system itself, to send a request (a “callout”) to an external application, or to analyze another piece of data. A rule takes the form of a logical expression that is compared against traffic and ultimately returns values of TRUE or FALSE.

The elements of the rule can themselves return TRUE or FALSE, string, or numeric values.

Before configuring a default syntax expression, you need to understand the characteristics of the data that the policy or other entity is to evaluate. For example, when working with the Integrated Caching feature, a policy determines what data can be stored in the cache. With Integrated Caching, you need to know the URLs, headers, and other data in the HTTP requests and responses that the NetScaler receives. With this knowledge, you can configure policies that match the actual data and enable the NetScaler to manage caching for HTTP traffic. This information helps you determine the type of expression that you need to configure in the policy.

Basic Elements of a Default Syntax Expression

A default syntax expression consists of, at a minimum, a prefix (or a single element used in place of a prefix). Most expressions also specify an operation to be performed on the data that the prefix identifies. You format an expression of up to 1,499 characters as follows:

```
<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]
```

where

<prefix>

is an anchor point for starting an expression.

The prefix is a period-delimited key that identifies a unit of data. For example, the following prefix examines HTTP requests for the presence of a header named Content-Type:

```
http.req.header("Content-Type")
```

Prefixes can also be used on their own to return the value of the object that the prefix identifies.

<operation>

identifies an evaluation that is to be performed on the data identified by the prefix.

For example, consider the following expression:

```
http.req.header("Content-Type").eq("text/html")
```

In this expression, the following is the operator component:

```
eq("text/html")
```

This operator causes the NetScaler to evaluate any HTTP requests that contain a Content-Type header, and in particular, to determine if the value of this header is equal to the string “text/html.” For more information, see Operations.

<compound-operator>

is a Boolean or arithmetic operator that forms a compound expression from multiple prefix or prefix.operation elements.

For example, consider the following expression:

```
http.req.header("Content-Type").eq("text/html") &&
http.req.url.contains(".html")
```

Prefixes

An expression prefix represents a discrete piece of data. For example, an expression prefix can represent an HTTP URL, an HTTP Cookie header, or a string in the body of an HTTP POST request. An expression prefix can identify and return a wide variety of data types, including the following:

- A client IP address in a TCP/IP packet
- NetScaler system time
- An external callout over HTTP
- A TCP or UDP record type

In most cases, an expression prefix begins with one of the following keywords:

CLIENT:

Identifies a characteristic of the client that is either sending a request or receiving a response, as in the following examples:

- The prefix `client.ip.dst` designates the destination IP address in the request or response.
- The prefix `client.ip.src` designates the source IP address.

HTTP:

Identifies an element in an HTTP request or a response, as in the following examples:

- The prefix `http.req.body(integer)` designates the body of the HTTP request as a multiline text object, up to the character position designated in integer.
- The prefix `http.req.header("header_name")` designates an HTTP header, as specified in header_name.
- The prefix `http.req.url` designates an HTTP URL in URL-encoded format.

SERVER:

Identifies an element in the server that is either processing a request or sending a response.

SYS:

Identifies a characteristic of the NetScaler that is processing the traffic.

Note: Note that DNS policies support only SYS, CLIENT, and SERVER objects.

In addition, in the Access Gateway, the Clientless VPN function can use the following types of prefixes:

TEXT:

Identifies any text element in a request or a response.

TARGET:

Identifies the target of a connection.

URL:

Identifies an element in the URL portion of an HTTP request or response.

As a general rule of thumb, any expression prefix can be a self-contained expression. For example, the following prefix is a complete expression that returns the contents of the HTTP header specified in the string argument (enclosed in quotation marks):

```
http.res.header.("myheader")
```

Or you can combine prefixes with simple operations to determine TRUE and FALSE values. For example, the following returns a value of TRUE or FALSE:

```
http.res.header.("myheader").exists
```

You can also use complex operations on individual prefixes and multiple prefixes within an expression, as in the following example:

```
http.req.url.length + http.req.cookie.length <= 500
```

Which expression prefixes you can specify depends on the NetScaler feature. The following table describes the expression prefixes that are of interest on a per-feature basis

Table 1. Permitted Types of Expression Prefixes in Various NetScaler Features

Feature	Types of Expression Prefix Used in the Feature
DNS	`SYS, CLIENT, SERVER`
Responder in Protection Features	`HTTP, SYS, CLIENT`
Content Switching	`HTTP, SYS, CLIENT`
Rewrite	`HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN`
Integrated Caching	`HTTP, SYS, CLIENT, SERVER`
Access Gateway, Clientless Access	`HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN`

Note: For details on the permitted expression prefixes in a feature, see the documentation for that feature.

Single-Element Expressions

The simplest type of default syntax expression contains a single element. This element can be one of the following:

- `true`. A default syntax expression can consist simply of the value `true`. This type of expression always returns a value of TRUE. It is useful for chaining policy actions and triggering Goto expressions.
- `false`. A default syntax expression can consist simply of the value `false`. This type of expression always returns a value of FALSE.
- A prefix for a compound expression. For example, the prefix `HTTP.REQ.HOSTNAME` is a complete expression that returns a host name and `HTTP.REQ.URL` is a complete expression that returns a URL. The prefix could also be used in conjuction with operations and additional prefixes to form a compound expression.

Operations

In most expressions, you also specify an operation on the data that the prefix identifies. For example, suppose that you specify the following prefix:

```
http.req.url
```

This prefix extracts URLs in HTTP requests. This expression prefix does not require any operators to be used in an expression. However, when you configure an expression that processes HTTP request URLs, you can specify operations that analyze particular characteristics of the URL. Following are a few possibilities:

- Search for a particular host name in the URL.
- Search for a particular path in the URL.
- Evaluate the length of the URL.
- Search for a string in the URL that indicates a time stamp and convert it to GMT.

The following is an example of a prefix that identifies an HTTP header named Server and an operation that searches for the string IIS in the header value:

```
http.res.header("Server").contains("IIS")
```

Following is an example of a prefix that identifies host names and an operation that searches for the string “www.mycompany.com” as the value of the name:

```
http.req.hostname.eq("www.mycompany.com")
```

Basic Operations on Expression Prefixes

The following table describes a few of the basic operations that can be performed on expression prefixes.

Table 1. Basic Operations for Expressions

Operation	Determines Whether or Not
`CONTAINS(<string>)`	The object matches <string>. Following is an example: `http.req.header("Cache-Control").contains("no-cache")`
`EXISTS`	A particular item is present in an object. Following is an example: `http.res.header("MyHdr").exists`
`EQ(<text>)`	A particular non-numeric value is present in an object. Following is an example: `http.req.method.eq(post)`
`EQ(<integer>)`	A particular numeric value is present in an object. Following is an example: `client.ip.dst.eq(10.100.10.100)`
`LT(<integer>)`	An object's value is less than a particular value. Following is an example: `http.req.content_length.lt(5000)`
`GT(<integer>)`	An object's value is greater than a particular value. Following is an example: `http.req.content_length.gt(5)`

The following table summarizes a few of the available types of operations.

Table 2. Basic Types of Operations

Operation Type	Description

Text operations	Match individual strings and sets of strings with any portion of a target. The target can be an entire string, the start of a string, or any portion of text in between the start and the end of the string. For example, you can extract the string "XYZ" from "XYZSomeText". Or, you can compare an HTTP header value with an array of different strings. You can also transform text into another type of data. Following are examples: • Transform a string into an integer value • Create a list from the query strings in a URL • Transform a string into a time value
Numeric operations	Numeric operations include applying arithmetic operators, evaluating content length, the number of items in a list, dates, times, and IP addresses.

Compound Default Syntax Expressions

You can configure a default syntax expression that contains Boolean or arithmetic operators and multiple atomic operations. The following compound expression contains a boolean AND:

```
http.req.hostname.eq("mycompany.com") && http.req.method.eq(post)
```

The following expression adds the value of two targets, and compares the result to a third value:

```
http.req.url.length + http.req.cookie.length <= 500
```

A compound expression can contain any number of logical and arithmetic operators. The following expression evaluates the length of an HTTP request on the basis of its URL and cookie, evaluates text in the header, and performs a Boolean AND on these two results:

```
http.req.url.length + http.req.cookie.length <= 500 &&
http.req.header.contains("some text")
```

You can use parentheses to control the order of evaluation in a compound expression.

Booleans in Compound Expressions

You configure compound expressions with the following operators:

&&.

This operator is a logical AND. For the expression to evaluate to TRUE, all components that are joined by the And must evaluate to TRUE. Following is an example:

```
http.req.url.hostname.eq("myHost") &&
http.req.header("myHeader").exists
```

||.

This operator is a logical OR. If any component of the expression that is joined by the OR evaluates to TRUE, the entire expression is TRUE.

!.

Performs a logical NOT on the expression.

In some cases, the NetScaler configuration utility offers AND, NOT, and OR operators in the Add Expression dialog box. However, these are of limited use. Citrix recommends that you use the operators &&, ||, and ! to configure compound expressions that use Boolean logic.

Parentheses in Compound Expressions

You can use parentheses to control the order of evaluation of an expression. The following is an example:

```
http.req.url.contains("myCompany.com") ||
(http.req.url.hostname.eq("myHost") &&
http.req.header("myHeader").exists)
```

The following is another example:

```
(http.req.header("Content-Type").exists &&
http.req.header("Content-Type").eq("text/html")) ||
(http.req.header("Transfer-Encoding").exists ||
http.req.header("Content-Length").exists)
```

Compound Operations for Strings

The following table describes operators that you can use to configure compound operations on string data.

Table 1. String-Based Operations for Compound Default Syntax Expressions

All string operations	
Operations that produce a string value	
str + str	Concatenates the value of the expression on the left of the operator with the value on the right. Following is an example: `http.req.hostname + http.req.url.protocol`
str + num	Concatenates the value of the expression on the left of the operator with a numeric value on the right. Following is an example: `http.req.hostname + http.req.url.content_length`
num + str	Concatenates the numeric value of the expression on the left side of the operator with a string value on the right. Following is an example: `http.req.url.content_length + http.req.url.hostname`
str + ip	Concatenates the string value of the expression on the left side of the operator with an IP address value on the right. Following is an example: `http.req.hostname + 10.00.000.00`
ip + str	Concatenates the IP address value of the expression on the left of the operator with a string value on the right.Following is an example: `client.ip.dst + http.req.url.hostname`
str1 ALT str2	Uses the string1 or string2 value that is derived from the expression on either side of the operator, as long as neither of these expressions is a compound expressions.Following is an example: `http.req.hostname alt client.ip.src`
Operations on strings that produce a result of TRUE or FALSE	
str == str	Evaluates whether the strings on either side of the operator are the same. Following is an example: `http.req.header("myheader") == http.res.header("myheader")`
str <= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or precedes it alphabetically.
str >= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or follows it alphabetically.

str < str	Evaluates whether the string on the left side of the operator precedes the string on the right alphabetically.
str > str	Evaluates whether the string on the left side of the operator follows the string on the right alphabetically.
str !!= str	Evaluates whether the strings on either side of the operator are different.
Logical operations on strings	

bool && bool	This operator is a logical AND. When evaluating the components of the compound expression, all components that are joined by the AND must evaluate to TRUE. Following is an example:
	
	`http.req.method.eq(GET) &&`
	`http.req.url.query.contains("viewReport && my_pagelabel")`

bool || bool	This operator is a logical OR. When evaluating the components of the compound expression, if any component of the expression that is joined by the OR evaluates to TRUE, the entire expression is TRUE. Following is an example:
	
	`http.req.url.contains(".js") ||`
	`http.res.header.("Content-Type").contains("javascript")`

!bool	`Performs a logical NOT on the expression.`

Compound Operations for Numbers

You can configure compound numeric expressions. For example, the following expression returns a numeric value that is the sum of an HTTP header length and a URL length:

```
http.req.header.length + http.req.url.length
```

The following tables describes operators that you can use to configure compound expressions for numeric data.

Table 1. Arithmetic Operations on Numbers

Operator	Description
num + num	Add the value of the expression on the left of the operator to the value of the expression on the right. Following is an example: `http.req.content_length + http.req.url.length`
num – num	Subtract the value of the expression on the right of the operator from the value of the expression on the left.
num * num	Multiply the value of the expression on the left of the operator with the value of the expression on the right. Following is an example: `client.interface.rxthroughput * 9`
num / num	Divide the value of the expression on the left of the operator by the value of the expression on the right.
num % num	Calculate the modulo, or the numeric remainder on a division of the value of the expression on the left of the operator by the value of the expression on the right. For example, the values "15 mod 4" equals 3, and "12 mod 4" equals 0.
~number	Returns a number after applying a bitwise logical negation of the number. The following example assumes that numeric.expression returns 12 (binary 1100): `~numeric.expression.` The result of applying the ~ operator is -11 (a binary 1110011, 32 bits total with all ones to the left). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

number ^ number	Compares two bit patterns of equal length and performs an XOR operation on each pair of corresponding bits in each number argument, returning 1 if the bits are different, and 0 if they are the same. Returns a number after applying a bitwise XOR to the integer argument and the current number value. If the values in the bitwise comparison are the same, the returned value is a 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): `numeric.expression1 ^ numeric.expression2` The result of applying the ^ operator to the entire expression is 6 (binary 0110). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
number | number	Returns a number after applying a bitwise OR to the number values. If either value in the bitwise comparison is a 1, the returned value is a 1. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): `numeric.expression1 | numeric.expression2` The result of applying the | operator to the entire expression is 14 (binary 1110). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
number & number	Compares two bit patterns of equal length and performs a bitwise AND operation on each pair of corresponding bits, returning 1 if both of the bits contains a value of 1, and 0 if either bits are 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): `numeric.expression1 & numeric.expression2` The whole expression evaluates to 8 (binary 1000). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

num << num	Returns a number after a bitwise left shift of the number value by the right-side number argument number of bits. Note that the number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: `numeric.expression1 << numeric.expression2` The result of applying the LSHIFT operator is 96 (a binary 1100000). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
num >> num	Returns a number after a bitwise right shift of the number value by the integer argument number of bits. Note that the number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: `numeric.expression1 >> numeric.expression2` The result of applying the RSHIFT operator is 1 (a binary 0001). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Table 2. Numeric Operators That Produce a Result of TRUE or FALSE

Operator	Description
num == num	Determine if the value of the expression on the left of the operator is equal to the value of the expression on the right.
num != num	Determine if the value of the expression on the left of the operator is not equal to the value of the expression on the right.
num > num	Determine if the value of the expression on the left of the operator is greater than the value of the expression on the right.
num < num	Determine if the value of the expression on the left of the operator is less than the value of the expression on the right.
num >= num	Determine if the value of the expression on the left of the operator is greater than or equal to the value of the expression on the right.
num <= num	Determine if the value of the expression on the left of the operator is less than or equal to the value of the expression on the right

Functions for Data Types in the Policy Infrastructure

The NetScaler policy infrastructure supports the following numeric data types:

- Integer (32 bits)
- Unsigned long (64 bits)
- Double (64 bits)

Simple expressions can return all of these data types. Therefore, you can create compound expressions that use arithmetic operators and logical operators to evaluate or return values of these data types. Additionally, you can use all of these values in policy expressions. Literal constants of type unsigned long can be specified by appending the string `ul` to the number. Literal constants of type double contain a period (.), an exponent, or both.

Arithmetic Operators, Logical Operators, and Type Promotion

In compound expressions, the following standard arithmetic and logical operators can be used for the double and unsigned long data types:

- `+`, `-`, `*`, and `/`
- `%`, `~`, `^`, `&`, `|`, `<<`, and `>>` (do not apply to double)
- `==`, `!=`, `>`, `<`, `>=`, and `<=`

All of these operators have the same meaning as in the C programming language.

In all cases of mixed operations between operands of type integer, unsigned long, and double, type promotion is performed so that the operation can be performed on operands of the same type. A type of lower precedence is automatically promoted to the type of the operand with the highest precedence involved in the operation. The order of precedence (higher to lower) is as follows:

- Double
- Unsigned long
- Integer

Therefore, an operation that returns a numeric result returns a result of the highest type involved in the operation.

For example, if the operands are of type integer and unsigned long, the integer operand is automatically converted to type unsigned long. This type conversion is performed even in simple expressions in which the type of data identified by the expression prefix does not match the type of data that is passed as the argument to the function. To illustrate such an example, in the operation `HTTP.REQ.CONTENT_LENGTH.DIV(3ul)`, the integer returned

by the prefix HTTP.REQ.CONTENT_LENGTH is automatically converted to unsigned long (the type of the data passed as the argument to the DIV() function), and an unsigned long division is performed. Similarly, the argument can be promoted in an expression. For example, HTTP.REQ.HEADER("myHeader").TYPECAST_DOUBLE_AT.DIV(5) promotes the integer 5 to type double and performs double-precision division.

The following table describes the arithmetic and Boolean functions that can be used with the integer, unsigned long, and double data types. For information about expressions for casting data of one type to data of another type, see Typecasting Data.

Function	Description
<prefix>.ADD(<integer>|<unsigned long>|<double>)	Adds the argument to the value of the expression prefix and returns the result **Example:** http.req.content_length.add(10)
<prefix>.SUB(<integer>|<unsigned long>|<double>)	Subtracts the argument from the value of the prefix and returns the result. **Example:** http.req.header.length.sub(10)
<prefix>.DIV(<integer>|<unsigned long>|<double>)	Divides the value of the prefix by the argument and returns the quotient. **Example:** http.req.content_length.div(2)
<prefix>.MUL(<integer>|<unsigned long>|<double>)	Multiplies the value of the prefix by the argument and returns the product. **Example:** http.req.content_length.mul(2)
<prefix>.BETWEEN(<lower_integer>, <higher_integer>|<lower_unsigned_long, higher_unsigned_long>|<lower_double>, <higher_double>)	Returns a Boolean TRUE if the value of the prefix is greater than or equal to t value argument and less than or equal to the higher value argument. **Example:** http.req.content_length.between(5, 500)
<prefix>.EQ(<integer>|<unsigned long>|<double>)	Returns a Boolean TRUE if the value of the prefix is equal to the argument. **Example:** http.req.content_length.eq(50)
<prefix>.NE(<integer>|<unsigned long>|<double>)	Returns a Boolean TRUE if the value of the prefix is not equal to the argumen **Example:** http.req.content_length.ne(50)

<prefix>.GE(<integer>|<unsigned long>|<double>)	Returns a Boolean TRUE if the value of the prefix is greater than or equal to t argument. **Example:** http.req.content_length.ge(500)
<prefix>.GT(<integer>|<unsigned long>|<double>)	Returns a Boolean TRUE if the value of the prefix is greater than the argumen **Example:** http.req.content_length.gt(500)
<prefix>.LE(<integer>|<unsigned long>|<double>)	Returns a Boolean TRUE if the value of the prefix is less than or equal to the argument. **Example:** http.req.content_length.le(5)
<prefix>.LT(<integer>|<unsigned long>|<double>)	Returns a Boolean TRUE if the value of the prefix is less than the argument. **Example:** http.req.content_length.lt(5)
<prefix>.NEG	Returns the negative of the value of the prefix. This function cannot be used v prefix that returns data of type unsigned long. **Example:** http.req.content_length.neg If the content length is 30 characters, the NEG function in the above example value of ▫30.

<prefix>.BITAND(<integer>|<unsigned long>)	Returns the result of a bitwise AND operation performed on the binary equiva the argument and the value returned by the prefix. The bitwise AND operation operates on each pair of corresponding bits in the bit strings. The operation returns 1 only if both bits are equal to 1. If either bi if both bits are 0, the operation returns 0. If the binary equivalent of an opera contains fewer than 32 bits, the function implicitly adds leading zeros to make operand 32 bits wide before performing the operation. The BITAND function be used with the double data type. **Example:** `http.req.header (\"test\").contains_index(\"patternset1\").bitand(4)` In the above example, assume that the index returned by the CONTAINS_IND pattern set function is an integer value of 12. The BITAND function performs AND operation between the binary value of 12, which is 00000000000000000000000000001100 (32 bits wide) and the binary value of 4, 00000000000000000000000000001100 (32 bits wide). The resulting bit string th function returns is 00000000000000000000000000000100, whose decimal equiv 4. An ampersand (&) performs a similar function to BITAND but takes two expre operands rather than an expression (the prefix) and the argument.
<prefix>.BITNEG	Returns the value that results from a bitwise negation of the value of the pref data type of the value that is returned is the same as that of the value that wo otherwise be returned by the prefix. This function cannot be used with a prefi returns data of type double. If the binary equivalent of an operand contains fe 32 bits, the function implicitly adds leading zeros to make the operand 32 bits before performing the operation. **Example:** `http.req.header(\"test\").contains_index(\"patternset1\").b` In the above example, assume that the index returned by the CONTAINS_IND pattern set function is an integer value of 12, whose binary value is 00000000000000000000000000001100 (32 bits wide). The BITNEG function ret 11111111111111111111111111110011, which represents an integer value of - 1 A tilde (~) performs a similar function to that of BITNEG but takes another ex as an argument, instead of operating on an integer prefix expression.

<prefix>.BITOR(<integer>|<unsigned long>)	Returns the result of a bitwise OR operation performed on the value of the pr the argument. The function returns 1 if either or both bits in a corresponding set to 1. If both bits are 0, the function returns 0. The BITOR function cannot with the double data type. If the binary equivalent of an operand contains few 32 bits, the function implicitly adds leading zeros to make the operand 32 bits before performing the operation. **Example:** `http.req.header (\"test\").contains_index(\"patternset1\").bitor(7)` In the above example, assume that the index returned by the CONTAINS_IND pattern set function is an integer value of 9. The BITOR function performs a bi operation on the binary value of 9, which is 0000000000000000000000000001 bits wide), and the binary value of 7, which is 0000000000000000000000000000 bits wide). The function returns 00000000000000000000000000001111, which represents an integer value of 15. The pipe (|) performs a similar function to that of BITOR but takes two expre operands rather than an integer or unsigned long (the argument to the functio expression prefix.
<prefix>.BITXOR(<integer>|<unsigned long>)	Returns the result of a bitwise EXLUSIVE-OR (XOR) operation performed on t of the prefix and the value of the argument. If the values of a pair of correspo bits are the same, the function returns 0. If the bits do not have the same valu function returns 1. If the binary equivalent of an operand contains fewer than the function implicitly adds leading zeros to make the operand 32 bits wide be performing the operation. The BITXOR function cannot be used with the doub type. **Example:** `http.req.header (\"test\").contains_index(\"patternset1\").bitxor(8)` In the above example, assume that the index returned by the CONTAINS_IND pattern set function is an integer value of 15. The BITOR function performs a XOR operation on the binary value of 15, which is 0000000000000000000000000 (32 bits wide), and the binary value of 8, which is 0000000000000000000000000 (32 bits wide). The function returns 00000000000000000000000000000111, whi represents an integer value of 7. A caret (^) performs a similar function to that of BITXOR but takes two expre operands rather than an expression and an argument.

<prefix>.LSHIFT(<integer>|<unsigned long>)	Returns the result of a bitwise left shift operation on the value of the prefix. T number of shifts is <integer> modulo 32. Each leftward shift effectively mu the value of the prefix by 2. If the binary equivalent of an operand contains fe 32 bits, the function implicitly adds leading zeros to make the operand 32 bits before performing the operation. This function cannot be used with a prefix th returns data of type double. **Example:** `http.req.header (\"test\").contains_index(\"pat1\").lshift(` Assume that the index that is returned by the CONTAINS_INDEX operator is 1 left shift operator drops the two leftmost bits in the binary value of 10, which 00000000000000000000000000001010 (32 bits wide), and adds two zeros to the The result is 00000000000000000000000000101000, which represents a decima 40. A double less-than (<<) performs a similar function to that of LSHIFT but tak expressions as operands, instead of an expression and an argument.
<prefix>.RSHIFT(<integer>|<unsigned long>)	Returns the result of a bitwise right shift operation on the value of the prefix. number of shifts is <integer> modulo 32. Each rightward shift effectively di value of the prefix by 2. If the binary equivalent of an operand contains less th bits, the function implicitly adds leading zeros before performing the operatio make the operand 32 bits wide. This function cannot be used with a prefix tha data of type double. **Example:** `http.req.header (\"test\").contains_index(\"pat1\").Rshift(` Assume that the index that is returned by the CONTAINS_INDEX operator is 3 left shift operator drops the two rightmost bits in the binary value of 320, whi 00000000000000000000000101000000 (32 bits wide), and adds two zeros to the The result is 00000000000000000000000001010000, which represents an intege of 80. A double greater-than (>>) performs the same function as RSHIFT but takes t expressions as operands, instead of an expression and an argument.

Specifying the Character Set in Expressions

The policy infrastructure on the Citrix® NetScaler® appliance supports the ASCII and UTF-8 character sets. The default character set is ASCII. If the traffic for which you are configuring an expression consists of only ASCII characters, you need not specify the character set in the expression. However, you must specify the character set in every simple expression that is meant for UTF-8 traffic. To specify the UTF-8 character set in a simple expression, you must include the `SET_CHAR_SET(<charset>)` function, with `<charset>` specified as UTF_8, as shown in the following examples:

```
HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8).CONTAINS("ß")

HTTP.RES.BODY(100).SET_CHAR_SET(UTF_8).BEFORE_STR("Bücher").AFTER_STR("Wörterbuch"
```

In an expression, the `SET_CHAR_SET()` function must be introduced at the point in the expression after which data processing must be carried out in the specified character set. For example, in the expression `HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding example/).CONTAINS_ANY("Greek_ alphabet")`, if the strings stored in the pattern set "Greek_alphabet" are in UTF-8, you must include the `SET_CHAR_SET(UTF_8)` function immediately before the `CONTAINS_ANY("<string>")` function, as follows:

```
HTTP.REQ.BODY(1000).AFTER_REGEX(re/following
example/).BEFORE_REGEX(re/In the preceding
example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("Greek_ alphabet")
```

The `SET_CHAR_SET()` function sets the character set for all further processing (that is, for all subsequent functions) in the expression unless it is overridden later in the expression by another `SET_CHAR_SET()` function that changes the character set. Therefore, if all the functions in a given simple expression are intended for UTF-8, you can include the `SET_CHAR_SET(UTF_8)` function immediately after functions that identify text (for example, the `HEADER("<name>")` or `BODY(<int>)` funtions). In the second example that follows the first paragraph above, if the ASCII arguments passed to the `AFTER_REGEX()` and `BEFORE_REGEX()` functions are changed to UTF-8 strings, you can include the `SET_CHAR_SET(UTF_8)` function immediately after the `BODY(1000)` function, as follows:

```
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).BEFORE_REGEX(re/W
```

The UTF-8 character set is a superset of the ASCII character set, so expressions configured for the ASCII character set continue to work as expected if you change the character set to UTF-8.

Compound Expressions with Different Character Sets

In a compound expression, if one subset of expressions is configured to work with data in the ASCII character set and the rest of the expressions are configured to work with data in the UTF-8 character set, the character set specified for each individual expression is considered when the expressions are evaluated individually. However, when processing the compound expression, just before processing the operators, the appliance promotes the character set of the returned ASCII values to UTF-8. For example, in the following compound expression, the first simple expression evaluates data in the ASCII character set while the second simple expression evaluates data in the UTF-8 character set:

```
HTTP.REQ.HEADER("MyHeader") == HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

However, when processing the compound expression, just before evaluating the "is equal to" Boolean operator, the NetScaler appliance promotes the character set of the value returned by `HTTP.REQ.HEADER("MyHeader")` to UTF-8.

The first simple expression in the following example evaluates data in the ASCII character set. However, when the NetScaler appliance processes the compound expression, just before concatenating the results of the two simple expressions, the appliance promotes the character set of the value returned by `HTTP.REQ.BODY(10)` to UTF-8.

```
HTTP.REQ.BODY(10) + HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

Consequently, the compound expression returns data in the UTF-8 character set.

Specifying the Character Set Based on the Character Set of Traffic

You can set the character set to UTF-8 on the basis of traffic characteristics. If you are not sure whether the character set of the traffic being evaluated is UTF-8, you can configure a compound expression in which the first expression checks for UTF-8 traffic and subsequent expressions set the character set to UTF-8. Following is an example of a compound expression that first checks the value of "charset" in the request's Content-Type header for "UTF-8" before checking whether the first 1000 bytes in the request contain the UTF-8 string "Bücher":

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T('=',
'; ', '"').VALUE("charset").EQ("UTF-8") &&
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).CONTAINS("Bücher")
```

If you are sure that the character set of the traffic being evaluated is UTF-8, the second expression in the example is sufficient.

Character and String Literals in Expressions

During expression evaluation, even if the current character set is ASCII, character literals and string literals, which are enclosed in single quotation marks (') and quotation marks (""), respectively, are considered to be literals in the UTF-8 character set. In a given expression, if a function is operating on character or string literals in the ASCII character set and you include a non-ASCII character in the literal, an error is returned.

Values in Hexadecimal and Octal Formats

When configuring an expression, you can enter values in octal and hexadecimal formats. However, each hexadecimal or octal byte is considered a UTF-8 byte. Invalid UTF-8 bytes result in errors regardless of whether the value is entered manually or pasted from the clipboard. For example, "\xce\x20" is an invalid UTF-8 character because "c8" cannot be followed by "20" (each byte in a multi-byte UTF-8 string must have the high bit set). Another example of an invalid UTF-8 character is "\xce \xa9," since the hexadecimal characters are separated by a white-space character.

Functions That Return UTF-8 Strings

Only the `<text>.XPATH` and `<text>.XPATH_JSON` functions always return UTF-8 strings. The following MYSQL routines determine at runtime which character set to return, depending on the data in the protocol:

- `MYSQL_CLIENT_T.USER`
- `MYSQL_CLIENT_T.DATABASE`
- `MYSQL_REQ_QUERY_T.COMMAND`
- `MYSQL_REQ_QUERY_T.TEXT`
- `MYSQL_REQ_QUERY_T.TEXT(<unsigned int>)`
- `MYSQL_RES_ERROR_T.SQLSTATE`
- `MYSQL_RES_ERROR_T.MESSAGE`
- `MYSQL_RES_FIELD_T.CATALOG`
- `MYSQL_RES_FIELD_T.DB`
- `MYSQL_RES_FIELD_T.TABLE`
- `MYSQL_RES_FIELD_T.ORIGINAL_TABLE`
- `MYSQL_RES_FIELD_T.NAME`
- `MYSQL_RES_FIELD_T.ORIGINAL_NAME`

- `MYSQL_RES_OK_T.MESSAGE`
- `MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned int>)`

Terminal Connection Settings for UTF-8

When you set up a connection to the NetScaler appliance by using a terminal connection (by using PuTTY, for example), you must set the character set for transmission of data to UTF-8.

Classic Expressions in Default Syntax Expressions

Classic expressions describe basic characteristics of traffic. In some cases, you may want to use a classic expression in a default syntax expression. You can do so with the default syntax expression configuration tool. This can be helpful when manually migrating the older classic expressions to the default syntax.

Note that when you upgrade the NetScaler to version 9.0 or higher, Integrated Caching policies are automatically upgraded to default syntax policies, and the expressions in these policies are upgraded to the default syntax.

The following is the syntax for all default syntax expressions that use a classic expression:

```
SYS.EVAL_CLASSIC_EXPR("expression")
```

Following are examples of the `SYS.EVAL_CLASSIC_EXPR("expression")` expression:

```
sys.eval_classic_expr("req.ssl.client.cipher.bits > 1000")
sys.eval_classic_expr("url contains abc")
sys.eval_classic_expr("req.ip.sourceip == 10.102.1.61 -netmask 255.255.255.255")
sys.eval_classic_expr("time >= *:30:00GMT")
sys.eval_classic_expr("e1 || e2")
sys.eval_classic_expr("req.http.urllen > 50")
sys.eval_classic_expr("dayofweek == wedGMT")
```

Configuring Default Syntax Expressions in a Policy

You can configure a default syntax expression of up to 1,499 characters in a policy. The user interface for default syntax expressions depends to some extent on the feature for which you are configuring the expression, and on whether you are configuring an expression for a policy or for another use.

When configuring expressions on the command line, you delimit the expression by using quotation marks (“. . .” or '. . .'). Within an expression, you escape additional quotation marks by using a back-slash (\). For example, the following are standard methods for escaping quotation marks in an expression:

```
"\"abc\""
```

```
‘\"abc\"’
```

You must also use a backslash to escape question marks and other backslashes on the command line. For example, the expression `http.req.url.contains(“\?”)` requires a backslash so that the question mark is parsed. Note that the backslash character will not appear on the command line after you type the question mark. On the other hand, if you escape a backslash (for example, in the expression `'http.req.url.contains("\\\\http")'`), the escape characters are echoed on the command line.

To make an entry more readable, you can escape the quotation marks for an entire expression. At the start of the expression you enter the escape sequence “q” plus one of the following special characters: `/{<|~$^+=&%@`?`.

You enter only the special character at the end of the expression, as follows:

```
q@http.req.url.contains("sometext") && http.req.cookie.exists@
```

```
q~http.req.url.contains("sometext") && http.req.cookie.exists~
```

Note that an expression that uses the { delimiter is closed with }.

For some features (for example, Integrated Caching and Responder), the policy configuration dialog box provides a secondary dialog box for configuring expressions. This dialog enables you to choose from drop-down lists that show the available choices at each point during expression configuration. You cannot use arithmetic operators when using these configuration dialogs, but most other default syntax expression features are available. To use arithmetic operators , write your expressions in free-form format.

To configure a default syntax rule by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a default syntax rule and verify the configuration:

1. `add cache|dns|rewrite|cs policy policyName` **`-rule`** `expression featureSpecificParameters` **`-action`**

2. `show cache|dns|rewrite|cs policy policyName`
   Following is an example of configuring a caching policy:

   Example

   > add cache policy pol-cache -rule http.req.content_length.le(5) -action INVAL
   Done

   > show cache policy pol-cache
   Name: pol-cache
   Rule: http.req.content_length.le(5)
   CacheAction: INVAL
   Invalidate groups: DEFAULT
   UndefAction: Use Global
   Hits: 0
   Undef Hits: 0

   Done

To configure a default syntax policy expression by using the configuration utility

1. In the navigation pane, click the name of the feature where you want to configure a policy, for example, you can select **Integrated Caching**, **Responder**, **DNS**, **Rewrite**, or **Content Switching**, and then click **Policies**.
2. Click **Add**.
3. For most features, click in the **Expression** field. For **Content Switching**, click **Configure**.
4. Click the **Prefix** icon (the house) and select the first expression prefix from the drop-down list. For example, in Responder, the options are **HTTP**, **SYS**, and **CLIENT**. The next set of applicable options appear in a drop-down list.
5. Double-click the next option to select it, and then type a period (.). Again, a set of applicable options appears in another drop-down list.
6. Continue selecting options until an entry field (signalled by parentheses) appears. When you see an entry field, enter an appropriate value in the parentheses. For example, if you select GT(int) (greater-than, integer format), you specify an integer in the parentheses. Text strings are delimited by quotation marks. Following is an example:

   HTTP.REQ.BODY(1000).BETWEEN("this","that")
7. To insert an operator between two parts of a compound expression, click the **Operators** icon (the sigma), and select the operator type. Following is an example of a configured expression with a Boolean OR (signalled by double vertical bars, ||):

   HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this","that")
8. To insert a named expression, click the down arrow next to the **Add** icon (the plus sign) and select a named expression.
9. To configure an expression using drop-down menus, and to insert built-in expressions, click the **Add** icon (the plus sign). The **Add Expression** dialog box works in a similar way to the main dialog box, but it provides drop-down lists for selecting options, and it provides text fields for data entry instead of parentheses. This dialog box also provides a Frequently Used Expressions drop-down list that inserts commonly used expressions. When you are done adding the expression, click **OK**.
10. When finished, click **Create**. A message in the status bar indicates that the policy expression is configured successfully.

To test a default syntax expression by using the configuration utility

1. In the navigation pane, click the name of the feature for which you want to configure a policy (for example, you can select **Integrated Caching**, **Responder**, **DNS**, **Rewrite**, or **Content Switching**), and then click **Policies**.
2. Select a policy and click **Open**.
3. To test the expression, click the **Evaluate** icon (the check mark).
4. In the expression evaluator dialog box, select the **Flow Type** that matches the expression.
5. In the **HTTP Request Data** or **HTTP Response Data** field, paste the HTTP request or response that you want to parse with the expression, and click **Evaluate**. Note that you must supply a complete HTTP request or response, and the header and body should be separated by blank line. Some programs that trap HTTP headers do not also trap the response. If you are copying and pasting only the header, insert a blank line at the end of the header to form a complete HTTP request or response.
6. Click **Close** to close this dialog box.

Configuring Named Default Syntax Expressions

Instead of retyping the same expression multiple times in multiple policies, you can configure a named expression and refer to the name any time you want to use the expression in a policy. For example, you could create the following named expressions:

ThisExpression:

```
http.req.body(100).contains("this")
```

ThatExpression:

```
http.req.body(100).contains("that")
```

You can then use these named expressions in a policy expression. For example, the following is a legal expression based on the preceding examples:

```
ThisExpression || ThatExpression
```

You can use the name of a default syntax expression as the prefix to a function. The named expression can be either a simple expression or a compound expression. The function must be one that can operate on the type of data that is returned by the named expression.

Example 1: Simple Named Expression as a Prefix

The following simple named expression, which identifies a text string, can be used as a prefix to the AFTER_STR("<string>")function, which works with text data:

```
HTTP.REQ.BODY(1000)
```

If the name of the expression is top1KB, you can use top1KB.AFTER_STR("username") instead of HTTP.REQ.BODY(1000).AFTER_STR("username").

Example 2: Compound Named Expression as a Prefix

You can create a compound named expression called basic_header_value to concatenate the user name in a request, a colon (:), and the user's password, as follows:

```
add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\"
+ HTTP.REQ.USER.PASSWD"
```

You can then use the name of the expression in a rewrite action, as shown in the following example:

```
add rewrite action insert_b64encoded_authorization insert_http_header
authorization '"Basic " + basic_header_value.b64encode'
-bypassSafetyCheck YES
```

In the example, in the expression that is used to construct the value of the custom header, the B64 encoding algorithm is applied to the string returned by the compound named expression.

You can also use a named expression (either by itself or as a prefix to a function) to create the text expression for the replacement target in a rewrite.

To configure a named default syntax expression by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure a named expression and verify the configuration:

- `add policy expression expressionName rule`
- `show policy expression expressionName`

  Example

  ```
  > add policy expression myExp "http.req.body(100).contains(\"the other\")"
   Done

  > show policy expression myExp
  1)      Name: myExp  Expr: "http.req.body(100).contains("the other")"  Hits: 0 Type : ADVANCED
   Done
  ```

  The expression can be up to 1,499 characters.

To configure a named expression by using the configuration utility

1. In the navigation pane, expand **AppExpert**, and then click **Expressions**.
2. Click **Advanced Expressions**.
3. Click **Add**.
4. Enter a name and a description for the expression.
5. Configure the expression by using the process described in To configure a default syntax policy expression by using the configuration utility. A message in the status bar indicates that the policy expression is configured successfully.

Configuring Default Syntax Expressions Outside the Context of a Policy

A number of functions, including the following, can require a default syntax expression that is not part of a policy:

Integrated Caching selectors:

You define multiple non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND relationship with the others.

Load Balancing:

You configure an expression for the TOKEN method of load balancing for a load balancing virtual server.

Rewrite actions:

Expressions define the location of the rewrite action and the type of rewriting to be performed, depending on the type of rewrite action that you are configuring. For example, a DELETE action only uses a target expression. A REPLACE action uses a target expression and an expression to configure the replacement text.

Rate-based policies:

You use default syntax expressions to configure Limit Selectors. You can use these selectors when configuring policies to throttle the rate of traffic to various servers. You define up to five non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND with the others.

To configure a default syntax expression outside a policy by using the NetScaler command line (cache selector example)

At the NetScaler command prompt, type the following commands to configure a default syntax expression outside a policy and verify the configuration:

- `add cache selector <selectorName> <expressions>`
- `show cache selector <selectorName>`

  Example

  ```
  > add cache selector mainpageSelector "http.req.cookie.value(\"ABC_def\")"  "http.req.url.query.value(\'
  selector "mainpageSelector" added
  ```

```
 Done
> show cache selector mainpageSelector
        Name: mainpageSelector
        Expressions:
                1) http.req.cookie.value("ABC_def")
                2) http.req.url.query.value("_ghi")
 Done
```

Following is an equivalent command that uses the more readable q delimiter, as described in Configuring Default Syntax Expressions in a Policy:

```
> add cache selector mainpageSelector2 q~http.req.cookie.value("ABC_def")~  q~http.req.url.query.valu
selector "mainpageSelector2" added
 Done
> show cache selector mainpageSelector2
        Name: mainpageSelector2
        Expressions:
                1) http.req.cookie.value("ABC_def")
                2) http.req.url.query.value("_ghi")
 Done
```

Default Syntax Expressions: Evaluating Text

You can configure a policy with a default syntax expression that evaluates text in a request or response. Default syntax text expressions can range from simple expressions that perform string matching in HTTP headers to complex expressions that encode and decode text. You can configure text expressions to be case sensitive or case insensitive and to use or ignore spaces. You can also configure complex text expressions by combining text expressions with Boolean operators

You can use expression prefixes and operators for evaluating HTTP requests, HTTP responses, and VPN and Clientless VPN data. However, text expression prefixes are not restricted to evaluating these elements of your traffic. For information about additional default syntax text expression prefixes and operators, see the following topics:

- Pattern Sets
- Regular Expressions
- Typecasting Data
- Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data
- Default Syntax Expressions: Parsing SSL Certificates
- Expressions for SSL Certificate Dates

About Text Expressions

You can configure various expressions for working with text that flows through the NetScaler appliance. Following are some examples of how you can parse text by using a default syntax expression:

- Determine that a particular HTTP header exists.

  For example, you may want to identify HTTP requests that contains a particular Accept-Language header for the purpose of directing the request to a particular server.

- Determine that a particular HTTP URL contains a particular string.

  For example, you may want to block requests for particular URLs. Note that the string can occur at the beginning, middle, or end of another string.

- Identify a POST request that is directed to a particular application.

  For example, you may want to identify all POST requests that are directed to a database application for the purpose of refreshing cached application data.

Note that there are specialized tools for viewing the data stream for HTTP requests and responses. For example, from the following URL, you can download a Firefox Web browser plug-in that displays HTTP request and response headers:

https://addons.mozilla.org/en-US/firefox/addon/3829

The following plug-in displays headers, query strings, POST data, and other information:

https://addons.mozilla.org/en-US/firefox/addon/6647

After you download these plug-ins, they are accessible from the Firefox Tools menu.

About Operations on Text

A text-based expression consists of at least one prefix to identify an element of data and usually (although not always) an operation on that prefix. Text-based operations can apply to any part of a request or a response. Basic operations on text include various types of string matches.

For example, the following expression compares a header value with a string:

```
http.req.header("myHeader").contains("some-text")
```

Following expressions are examples of matching a file type in a request:

```
http.req.url.suffix.contains("jpeg")

http.req.url.suffix.eq("jpeg")
```

In the preceding examples, the contains operator permits a partial match and the eq operator looks for an exact match.

Other operations are available to format the string before evaluating it. For example, you can use text operations to strip out quotes and white spaces, to convert the string to all lowercase, or to concatenate strings.

Note: Complex operations are available to perform matching based on patterns or to convert one type of text format to another type. For more information, see the following topics:

- Pattern Sets.
- Regular Expressions.
- Typecasting Data.

Compounding and Precedence in Text Expressions

You can apply various operators to combine text prefixes or expressions. For example, the following expression concatenates the returned values of each prefix:

```
http.req.hostname + http.req.url
```

Following is an example of a compound text expression that uses a logical AND. Both components of this expression must be TRUE for a request to match the expression:

```
http.req.method.eq(post) &&
http.req.body(1024).startswith("destination=")
```

Note: For more information on operators for compounding, see Compound Default Syntax Expressions.

Categories of Text Expressions

The primary categories of text expressions that you can configure are:

- Information in HTTP headers, HTTP URLs, and the POST body in HTTP requests.

  For more information, see Expression Prefixes for Text in HTTP Requests and Responses.

- Information regarding a VPN or a clientless VPN.

  For more information, see Expression Prefixes for VPNs and Clientless VPNs.

- TCP payload information.

  For more information about TCP payload expressions, see Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data.

- Text in a Secure Sockets Layer (SSL) certificate.

For information about text expressions for SSL and SSL certificate data, see Default Syntax Expressions: Parsing SSL Certificates and Expressions for SSL Certificate Dates.

Note: Parsing a document body, such as the body of a POST request, can affect performance. You may want to test the performance impact of policies that evaluate a document body.

Guidelines for Text Expressions

From a performance standpoint, it typically is best to use protocol-aware functions in an expression. For example, the following expression makes use of a protocol-aware function:

```
HTTP.REQ.URL.QUERY
```

The previous expression performs better than the following equivalent expression, which is based on string parsing:

```
HTTP.REQ.URL.AFTER_STR("?")
```

In the first case, the expression looks specifically at the URL query. In the second case, the expression scans the data for the first occurrence of a question mark.

There is also a performance benefit from structured parsing of text, as in the following expression:

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(For more information on typecasting, see Typecasting Data.) The typecasting expression, which collects comma-delimited data and structures it into a list, typically would perform better than the following unstructured equivalent:

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

Finally, unstructured text expressions typically have better performance than regular expressions. For example, the following is an unstructured text expression:

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```

The previous expression would generally provide better performance than the following equivalent, which uses a regular expression:

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

For more information on regular expressions, see Regular Expressions.

Expression Prefixes for Text in HTTP Requests and Responses

An HTTP request or response typically contains text, such as in the form of headers, header values, URLs, and POST body text. You can configure expressions to operate on one or more of these text-based items in an HTTP request or response.

The following table describes the expression prefixes that you can configure to extract text from different parts of an HTTP request or response

Table 1. HTTP Expression Prefixes That Return Text

	Description
`ODY(<integer>)`	Returns the body of an HTTP request as a multiline text object, up to the character positio <integer>. There is no maximum value for the body argument, but you should use as small a value as i values can affect performance. **Note:** Although it is possible to specify this prefix without an integer argument, this usage
`OSTNAME`	Returns the HTTP host name in the first line of the request, if there is one. Otherwise, this in the last occurrence of the HOST header. Note that there are two similar prefixes that return host names, as follows: • `http.req.url.hostname` only returns the host name from the URL • `http.req.header("Host")` only returns the value from the Host header. To use th you must typecast this string, as illustrated in the following example: `http.req.header("host").typecast_http_hostname_t` For more information on typecasting, see Typecasting Data..
`OSTNAME. DOMAIN`	Returns the domain name part of the host name. For example, if the host name is www.my www.myhost.com:8080, the domain is myhost.com. Returns incorrect results if the host name has an IP address. For information on expressions Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs. All text operations that you specify after this prefix are case insensitive.
`OSTNAME. SERVER`	Returns the server name part of the host name. If the host name is www.myhost.com or ww the server is www.myhost.com. All text operations that you specify after this prefix are case insensitive.

ETHOD	Returns the value of the METHOD in an HTTP request, or matches the method type if you p argument, for example, `http.req.method.eq(get)`. If you enclose the argument in quo sensitive.
RL	Returns the HTTP URL.
RL.HOSTNAME	Returns the host name in the HTTP URL. Do not use this prefix in bidirectional policies. Note that there are two similar prefixes that return host names, as follows: • `HTTP.REQ.HOSTNAME` returns the host name from the URL if there is one; otherwise, the last occurrence of the Host header. • `HTTP.REQ.HEADER("Host")` only returns the value from the Host header. To use th you must typecast this string, as illustrated in the following example: `http.req.header("host").typecast_http_hostname_t` For more information on typecasting, see Typecasting Data..
RL.HOSTNAME.DOMAIN	Returns the domain name part of the host name. For example, if the host name is www.my www.myhost.com:8080, the domain is myhost.com. This operation returns incorrect results if the host name has an IP address. For information addresses, see Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs. All text operations that you specify after this prefix are case insensitive unless explicitly se SET_TEXT_MODE operator.
RL.HOSTNAME.SERVER	Returns the server name part of the host name. For example, if the host name is www.myh www.myhost.com:8080, the server is www.myhost.com. All text operations that you specify after this prefix are case insensitive.
RL.PATH	Returns a slash- (/) separated list from the path in a URL. For example, if the URL is http://www.myhost.com/a/b/c/mypage.html?a=1, this prefix r /a/b/c/mypage.html. The expression `http.req.url.path.get(1)` returns "a" from the preceding URL. For m GET operation, see Expressions for Extracting Segments of URLs.
RL.PATH_AND_QUERY	Returns the portion of the URL that follows the host name. For example, if the URL is http://www.myhost.com/a/b/c/mypage.html?a=1, this prefix r /a/b/c/mypage.html?a=1.
RL.PROTOCOL	Returns the protocol in the URL. This prefix cannot be used in bidirectional policies. Following is an example: `http.req.hostname + http.req.url.protocol`

RL.QUERY	Returns a name-value list, using the delimiters “=” and “&” from the query component in t Following is an example: `http.req.url.query.contains("viewReport && my_pagelabel")`
RL.QUERY.VALUE	Returns the value from the name-value pair in the argument supplied to this prefix, using t the query component in the URL. Following is an example: `http.req.url.query.value("action")` The first component that matches the name is selected. The matching process honors the I NOIGNORECASE text modes. The URLENCODED and the NOURLENCODED text modes are igno
RL.SUFFIX	Returns the file name suffix in a URL. For example, if the path in the URL is /a/b/c/mypage.html, this suffix selects “html”. Foll example: `http.req.url.suffix.contains("jpeg")`
SER.NAME	Returns the name of the user in the request. Following is an example: `http.req.username.contains("rohit")`
ERSION	Returns the HTTP version listed in the request. Following is an example: `http.req.version "\"HTTP/1.0\"`
ODY(<integer>)	Returns a portion of the HTTP response body. The length of the returned text is equal to th <integer> argument. If there are fewer characters in the body than are specified in <integer>, the entire body is an example: `http.res.body(100).suffix('L',1)`
TATUS_MSG	Returns the HTTP response status message.
ERSION	Returns the HTTP version listed in the response.
RL.HOSTNAME.EQ(<hostname>)	Returns a Boolean TRUE value if the host name matches the <hostname> argument. The co insensitive and if textmode is URLENCODED, the host name is decoded before comparison. name is www.mycompany.com., the following is true: `http.req.url.hostname.eq("www.mycompany.com")`
RL.HOSTNAME.PORT	Returns the port in the host name. The string following and including the first colon (“:”) is value. For example, if the host name is www.mycompany.com:8080, the port is ":8080". If www.mycompany.com: the port is ":". If the host name is www.mycompany.com, the port location just after ".com". If the numerical value in the port is missing, it assumes a default value of 80 or 443 (for HT

RL.HOSTNAME.SERVER	Returns the server name portion of the host name. For example, if the host name is www.m www.mycompany.com:8080, the returned server name is www.mycompany.com. This method sets the text mode to case insensitive. All text operations after this method ar
RL.CVPN_ENCODE	Converts the URL to the clientless VPN format.

RL.PATH.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list. For example, if the element delimiter in the list is a
	list has an empty element following a=10:
	
	a=10,b=11, ,c=89
	
	The element following b=11 is not considered an empty element.
	
	As another example, consider the following header:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_ele`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_list_t(','). count`

RL.QUERY.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a name-value list. For example, if the list delim
	following list has an empty element following a=10:
	
	a=10;;b=11; ;c=89
	
	The element following b=11 is not considered an empty element.
	
	For example, consider the following header:
	
	Cust_Header : a=1;;b=2; ;c=3
	
	The following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore_emp`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';'). count`

Table 2. HTTP Expression Prefixes That Return Text

	Description
nteger>)	Returns the body of an HTTP request as a multiline text object, up to the character <integer>. There is no maximum value for the body argument, but you should use as small a va values can affect performance. **Note:** Although it is possible to specify this prefix without an integer argument, th

ME	Returns the HTTP host name in the first line of the request, if there is one. Otherwi in the last occurrence of the HOST header. Note that there are two similar prefixes that return host names, as follows: • `http.req.url.hostname` only returns the host name from the URL • `http.req.header("Host")` only returns the value from the Host header. To you must typecast this string, as illustrated in the following example: `http.req.header("host").typecast_http_hostname_t` For more information on typecasting, see Typecasting Data.
ME. DOMAIN	Returns the domain name part of the host name. For example, if the host name is w www.myhost.com:8080, the domain is myhost.com. Returns incorrect results if the host name has an IP address. For information on exp Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs. All text operations that you specify after this prefix are case insensitive.
ME. SERVER	Returns the server name part of the host name. If the host name is www.myhost.co the server is www.myhost.com. All text operations that you specify after this prefix are case insensitive.
	Returns the value of the METHOD in an HTTP request, or matches the method type argument, for example, `http.req.method.eq(get)`. If you enclose the argumen sensitive.
	Returns the HTTP URL.
STNAME	Returns the host name in the HTTP URL. Do not use this prefix in bidirectional policies. Note that there are two similar prefixes that return host names, as follows: • `HTTP.REQ.HOSTNAME` returns the host name from the URL if there is one; oth the last occurrence of the Host header. • `HTTP.REQ.HEADER("Host")` only returns the value from the Host header. To you must typecast this string, as illustrated in the following example: `http.req.header("host").typecast_http_hostname_t` For more information on typecasting, see Typecasting Data.
STNAME.DOMAIN	Returns the domain name part of the host name. For example, if the host name is w www.myhost.com:8080, the domain is myhost.com. This operation returns incorrect results if the host name has an IP address. For infor addresses, see Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN All text operations that you specify after this prefix are case insensitive unless expl SET_TEXT_MODE operator.

STNAME.SERVER	Returns the server name part of the host name. For example, if the host name is ww www.myhost.com:8080, the server is www.myhost.com. All text operations that you specify after this prefix are case insensitive.
TH	Returns a slash- (/) separated list from the path in a URL. For example, if the URL is http://www.myhost.com/a/b/c/mypage.html?a=1, this p /a/b/c/mypage.html. The expression `http.req.url.path.get(1)` returns "a" from the preceding URL GET operation, see Expressions for Extracting Segments of URLs.
TH_AND_QUERY	Returns the portion of the URL that follows the host name. For example, if the URL is http://www.myhost.com/a/b/c/mypage.html?a=1, this p /a/b/c/mypage.html?a=1.
OTOCOL	Returns the protocol in the URL. This prefix cannot be used in bidirectional policies. Following is an example: `http.req.hostname + http.req.url.protocol`
ERY	Returns a name-value list, using the delimiters “=” and “&” from the query compon Following is an example: `http.req.url.query.contains("viewReport && my_pagelabel")`
ERY.VALUE	Returns the value from the name-value pair in the argument supplied to this prefix, the query component in the URL. Following is an example: `http.req.url.query.value("action")` The first component that matches the name is selected. The matching process honc NOIGNORECASE text modes. The URLENCODED and the NOURLENCODED text modes
FFIX	Returns the file name suffix in a URL. For example, if the path in the URL is /a/b/c/mypage.html, this suffix selects “htm example: `http.req.url.suffix.contains("jpeg")`
	Returns the AAA user associated with the current HTTP transaction.
XTERNAL_GROUPS	Returns a list of the external groups to which a user belongs. The groups are separa For example, HTTP.REQ.USER.EXTERNAL_GROUPS returns a comma-separated list o which the user belongs.

XTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list of external groups to which the user belongs.
	
	If the element delimiter in the list is a comma (","), then the following list has an er
	
	a=10,,b=11, ,c=89
	
	But the element following "b=11" is not considered an empty element.
	
	For example, consider the following header in an HTTP request packet:
	
	Cust_Header : 123,,24, ,15
	
	Then the following expression returns a value of 4:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').  IGNORE_EMP`
	
	The following expression returns a value of 5:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').COUNT.`

XTERNAL_GROUPS(sep)	Returns a list of all the external groups to which the user belongs. The groups are so For example, the following expression gives a list of all the external groups, and the colon (":"): `HTTP.REQ.USER.EXTERNAL_GROUPS(':')` Parameters: sep - delimiter

XTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list of external groups to which the user belongs.
	
	If the element delimiter in the list is a comma (","), then the following list has an er
	
	a=10,,b=11, ,c=89
	
	But the element following "b=11" is not considered an empty element.
	
	For example, consider the following header in an HTTP request packet:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').  IGNORE_EMP`
	
	The following expression returns a value of 5:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').COUNT`

ROUPS	Returns a list of the internal and external groups to which the user belongs. The gro (","). In this list, internal groups are listed first, followed by external groups.

ROUPS.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list of groups to which the user belongs.
	
	If the element delimiter in the list is a comma (","), then the following list has an er
	
	a=10,,b=11, ,c=89
	
	But the element that follows "b=11" is not considered an empty element.
	
	For example, consider the following header in an HTTP request packet:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(','). IGNORE_EMP`
	
	The following expression returns a value of 5:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').COUNT`

ROUPS(sep)	Returns a list of groups to which the user belongs. The groups in the list are separat the argument. For example, the following expression returns a colon-separated list of all the group `HTTP.REQ.USER.GROUPS(':')` In this list, internal groups are listed first, followed by external groups. Parameters: sep - delimiter

ROUPS.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list of groups to which the user belongs.
	
	If the element delimiter in the list is a comma (","), then the following list has an er
	
	a=10,,b=11, ,c=89
	
	But the element following "b=11" is not considered an empty element.
	
	For example, consider the following header in an HTTP request packet:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(','). IGNORE_EMP`
	
	The following expression returns a value of 5:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').COUNT.`

NTERNAL_GROUPS	Returns a list of internal groups to which the user belongs. The groups are separate For example, the following expression returns a comma-separated list of all the inte belongs. `HTTP.REQ.USER.INTERNAL_GROUPS`

NTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list of internal groups to which the user belongs.
	
	If the element delimiter in the list is a comma (","), then the following list has an er
	
	a=10,,b=11, ,c=89
	
	But the element following "b=11" is not considered an empty element.
	
	For example, consider the following header in an HTTP request packet:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(','). IGNORE_EMP`
	
	The following expression returns a value of 5:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').COUNT`

NTERNAL_GROUPS(sep)	Returns a list of the internal groups to which the user belongs. The groups are separ For example, the following expression returns a colon-separated list of all the inter belongs. `HTTP.REQ.USER.INTERNAL_GROUPS(':')` Parameters: sep - delimiter

NTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list of internal groups to which the user belongs.
	
	If the element delimiter in the list is a comma (","), then the following list has an er
	
	a=10,,b=11, ,c=89
	
	But the element following "b=11" is not considered an empty element.
	
	For example, consider the following header in an HTTP request packet:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(','). IGNORE_EMP`
	
	The following expression returns a value of 5:
	
	`HTTP.REQ.HEADER("Cust_Header").TYPECAST_LIST_T(',').COUNT.`

S_MEMBER_OF(group_name)	Returns a boolean TRUE if the user who is named in the request is a member of the Following is an example: `http.req.user.is_member_of("mygroup")` Parameter: group_name: The name of the group.
AME	Returns the name of the user in the request. Following is an example: `http.req.username.contains("rohit")`
ASSWD	Returns the password of the user.
N	Returns the HTTP version listed in the request. Following is an example: `http.req.version "\"HTTP/1.0\"`
Integer>)	Returns a portion of the HTTP response body. The length of the returned text is equ <integer> argument. If there are fewer characters in the body than are specified in <integer>, the entire an example: `http.res.body(100).suffix('L',1)`
_MSG	Returns the HTTP response status message.
N	Returns the HTTP version listed in the response.
STNAME.EQ(<hostname>)	Returns a Boolean TRUE value if the host name matches the <hostname> argument. insensitive and if textmode is URLENCODED, the host name is decoded before comp name is www.mycompany.com., the following is true: `http.req.url.hostname.eq("www.mycompany.com")`
STNAME.PORT	Returns the port in the host name. The string following and including the first colon value. For example, if the host name is www.mycompany.com:8080, the port is ":80 www.mycompany.com: the port is ":". If the host name is www.mycompany.com, th location just after ".com". If the numerical value in the port is missing, it assumes a default value of 80 or 443
STNAME.SERVER	Returns the server name portion of the host name. For example, if the host name is www.mycompany.com:8080, the returned server name is www.mycompany.com. This method sets the text mode to case insensitive. All text operations after this me
M_OR_NEGOTIATE	Returns a Boolean TRUE if the request is a part of an NTLM or NEGOTIATE connectio
PN_ENCODE	Converts the URL to the clientless VPN format.

TH.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list. For example, if the element delimiter in the list has an empty element following a=10:
	
	a=10,b=11, ,c=89
	
	The element following b=11 is not considered an empty element.
	
	As another example, consider the following header:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_list_t(',').ignore_empt`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_list_t(','). count`

ERY.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a name-value list. For example, if the li following list has an empty element following a=10:
	
	a=10;;b=11; ;c=89
	
	The element following b=11 is not considered an empty element.
	
	For example, consider the following header:
	
	Cust_Header : a=1;;b=2; ;c=3
	
	The following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignor`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';'). cour`

Expression Prefixes for VPNs and Clientless VPNs

The default syntax expression engine provides prefixes that are specific to parsing VPN or Clientless VPN data. This data includes the following:

- Host names, domains, and URLs in VPN traffic.
- Protocols in the VPN traffic.
- Queries in the VPN traffic.

These text elements are often URLs and components of URLs. In addition to applying the text-based operations on these elements, you can parse these elements by using operations that are specific to parsing URLs. For more information, see Expressions for Extracting Segments of URLs.

The following table describes the expression prefixes for this type of data.

Table 1. VPN and Clientless VPN Expression Prefixes That Return Text

VPN Expression	Description
PN_DECODE	Extracts the original URL from a clientless VPN URL.
PN_ENCODE	Converts a URL to clientless VPN format.
STNAME	Extracts the HTTP host name from the host name in the URL. This prefix cannot be used in bidirectional policies.
STNAME.DOMAIN	Extracts the domain name from the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:80 mycompany.com. This prefix returns incorrect results if the host name is an IP address. For information addresses, see Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN All text operations after this prefix are case insensitive.
STNAME.EQ (<hostname>)	Returns a Boolean TRUE if the host name matches <hostname>. The comparison is ca For example, if the host name is www.mycompany.com, the following returns TRUE: `vpn.baseurl.hostname.eq("www.mycompany.com")` If the text mode is URLENCODED, the host name is decoded before comparison. For n Operations for HTTP, HTML, and XML Encoding and “Safe” Characters.

STNAME.SERVER	Evaluates the server portion of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:80 www.mycompany.com. All text operations after this prefix are case insensitive.
TH	Extracts a slash- (/) separated list from the path component of the URL. For example /a/b/c/mypage.html from the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 The following expression selects just the “a”: `http.req.url.path.get(1)` For more information on the GET operation, see Expressions for Extracting Segments

TH.IGNORE_EMPTY_ELEMENTS	This prefix ignores the elements in a list. For example, the following comma-separat
	after “a=10”:
	
	a=10,,b=11, ,c=89
	
	The element following b=11 contains a space, and by default, is not considered an er
	
	Consider the following HTTP header:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a count of 4 when evaluating this header:
	
	`http.req.header("Cust_Header").typecase_list_t(',').ignore_empty`
	
	The following expression returns a count of 5 when evaluating this header:
	
	`http.req.header("Cust_Header").typecase_list_t(','). count`

TH_AND_QUERY	Evaluates the text in the URL that follows the host name. For example, if the URL is http://www.mycompany.com/a/b/c/mypage.html?a=1, th /a/b/c/mypage.html?a=1.
OTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
ERY	Extracts a name-value list, using the “=” and “&” delimiters from the query string in

ERY.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a name-value list. For example, in the fo is an empty element following “a=10”:
	
	a=10;;b=11; ;c=89
	
	The element following b=11 contains a space and is not considered an empty elemen
	
	Consider the following HTTP header:
	
	Cust_Header : a=1;;b=2; ;c=3
	
	The following expression produces a count of 4 after evaluating this header:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore`
	
	The following expression produces a count of 5 after evaluating the header:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';').`

FFIX	Evaluates the file name suffix in a URL. For example, if the path is /a/b/c/my.page.html, this operation selects “html.”
_BASEURL	Evaluates the clientless VPN base URL.
_BASEURL.CVPN_DECODE	Extracts the original URL from the clientless VPN formatted URL.
_BASEURL.CVPN_ENCODE	Converts a URL to the clientless VPN format.
_BASEURL.HOSTNAME	Evaluates the host name in the URL. Do not use this prefix in bidirectional policies.
_BASEURL.HOSTNAME.DOMAIN	Evaluates the domain name part of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:80 mycompany.com. This operation returns incorrect results if the host name is an IP address. For informa addresses, see Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN All text operations after this prefix are case insensitive.
_BASEURL.HOSTNAME.EQ(<hostname>)	Returns a Boolean TRUE if the host name matches <hostname>. For example, if the host name is www.mycompany.com or www.mycompany.com:80 `vpn.clientless_baseurl. hostname.eq("www.mycompany.com")` The comparison is case insensitive. If the textmode is URLENCODED, the host name is For more information, see Operations for HTTP, HTML, and XML Encoding and “Safe”
_BASEURL.HOSTNAME.SERVER	Evaluates the server part of a host name. For example, if the host name is www.mycompany.com or www.mycompany.com:80 www.mycompany.com. All text operations after this prefix are case insensitive.

_BASEURL.PATH	Evaluates a slash- (/) separated list in the URL path. For example, this prefix selects /a/b/c/mypage.html from the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 The following expression selects “a” from the preceding URL: `http.req.url.path.get(1)` For more information on the GET operation, see Expressions for Extracting Segments

_BASEURL.PATH.IGNORE_EMPTY_ELEMENTS	Ignores empty elements in a list. For example, if the list delimiter is a comma (,) the element following “a=10”:
	
	a=10,b=11, ,c=89
	
	The element following b=11 contains a space and is not considered an empty elemen
	
	Consider the following HTTP header:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4 after evaluating this header:
	
	`http.req.header("Cust_Header").typecast_list_t(',').ignore_empty`
	
	The following expression returns a value of 5 after evaluating this header:
	
	`http.req.header("Cust_Header").typecast_list_t(',').`

_BASEURL.PATH_AND_QUERY	Evaluates the text following the host name in a URL. For example, this prefix selects /a/b/c/mypage.html?a=1 from the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1
_BASEURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
_BASEURL.QUERY	Extracts a name-value list that uses the delimiters “=” and “&” from a URL query str

_BASEURL.QUERY.IGNORE_EMPTY_ ELEMENTS	Ignores empty elements in a name-value list. For example, the following list contains “a=10”:
	
	a=10;;b=11; ;c=89
	
	The element following b=11 contains a space and is not considered an empty elemen
	
	As another example, consider the following http header:
	
	Cust_Header : a=1;;b=2; ;c=3
	
	The following expression returns a value of 4 after evaluating the preceding header:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore`
	
	The following expression returns a value of 5 after evaluating the preceding header:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';')`

_BASEURL.SUFFIX	Evaluates the file suffix in a URL. For example, if the URL path is /a/b/c/mypage.htm html.
_HOSTURL	Selects the clientless VPN host URL.
_HOSTURL.CVPN_DECODE	Selects the original URL from the clientless VPN formatted URL.
_HOSTURL.CVPN_ENCODE	Converts a URL to clientless VPN format.
_HOSTURL.HOSTNAME	Extracts the host name in the URL. Do not use this prefix in bidirectional policies.
_HOSTURL.HOSTNAME.DOMAIN	Extracts the domain name from the host name. For example, if the host name is www www.mycompany.com:8080, the domain is mycompany.com. This operation returns incorrect results if the host name contains an IP address. For i IP addresses, see Default Syntax Expressions: IP and MAC Addresses, Throughput, VLA All text operations after this prefix are case insensitive.
_HOSTURL.HOSTNAME.EQ(<hostname>)	Results in Boolean TRUE if the host name matches the <hostname> argument. The For example, if the host name is www.mycompany.com or www.mycompany.com., t TRUE: `vpn.clilentless_hosturl. hostname.eq("www.mycompany.com")` If the text mode is URLENCODED, the host name is decoded before comparison. For m Operations for HTTP, HTML, and XML Encoding and “Safe” Characters.
_HOSTURL.HOSTNAME.SERVER	Evaluates the server part of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:80 www.mycompany.com. The comparison is case insensitive, and all text operations after this method are case

_HOSTURL.PATH	Evaluates a slash- (/) separated list on the path component of the URL. For example, consider the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 This prefix selects /a/b/c/mypage.html from the preceding URL.

_HOSTURL.PATH.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a list. For example, if the delimiter in a l contains an empty element after the entry “a=10”:
	
	a=10,b=11, ,c=89
	
	The element following b=11 contains a space and is not considered an empty elemen
	
	Consider the following header:
	
	Cust_Header : 123,,24, ,15
	
	The following expression returns a value of 4 for this header:
	
	`http.req.header("Cust_Header").typecast_list_t(','). ignore_empt`
	
	The following expression returns a value of 5 for the same header:
	
	`http.req.header("Cust_Header").typecast_list_t(',').`

_HOSTURL.PATH_AND_QUERY	Evaluates the portion of the URL that follows the host name. For example, consider the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 This prefix returns /a/b/c/mypage.html?a=1 from the preceding URL.
_HOSTURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
_HOSTURL.QUERY	Extracts a name-value list, using the “=” and “&” delimiters from a URL query string.

_HOSTURL.QUERY.IGNORE_EMPTY_ ELEMENTS	Ignores empty elements in a name-value list. For example, the following list uses a s
	contains an empty element after “a=10”:
	
	a=10;;b=11; ;c=89
	
	In the preceding example, the element following b=11 is not considered an empty ele
	
	Consider the following header:
	
	Cust_Header : a=1;;b=2; ;c=3
	
	The following expression returns a value of 4 after evaluating this header:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore`
	
	The following expression returns a value of 5 after evaluating the same header:
	
	`http.req.header("Cust_Header").typecast_nvlist_t('=',';')`

_HOSTURL.SUFFIX	Extracts a file name suffix in a URL. For example, if the path is /a/b/c/my.page.html, this prefix selects html.
N	Extracts the domain name part of the host name. For example, if the host name is w www.mycompany.com:8080, the domain is mycompany.com. This prefix returns incorrect results if the host name contains an IP address. For infor addresses, see Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN All text operations after this prefix case insensitive.
ostname>)	Returns a Boolean TRUE value if the host name matches the <hostname>. The compa For example, if the host name is www.mycompany.com or www.mycompany.com:80 TRUE: `vpn.host.eq("www.mycompany.com")` If the text mode is URLENCODED the host name is decoded before comparison. For m Operations for HTTP, HTML, and XML Encoding and “Safe” Characters.
R	Extracts the server name part of the host name. For example, if the host name is ww www.mycompany.com:8080, the server is www.mycompany.com. All text operations after this prefix are case insensitive.

Basic Operations on Text

Basic operations on text include operations for string matching, calculating the length of a string, and controlling case sensitivity. You can include white space in a string that is passed as an argument to an expression, but the string cannot exceed 255 characters.

Identifying Requests or Responses that Match or Contain a Given String

The following table lists basic string matching operations in which the functions return a Boolean TRUE or FALSE.

Table 1. Functions for Identifying Requests or Responses that Match or Contain a Given String

Function	Description
`<text>.CONTAINS(<string>)`	Returns a Boolean TRUE value if the target contains <string>. Following is an example: `http.req.url.contains(".jpeg")`
`<text>.EQ(<string>)`	Returns a Boolean TRUE value if the target is an exact match with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of “myhostabc”: `http.req.url.hostname.eq("myhostabc")`
`<text>.STARTSWITH(<string>)`	Returns a Boolean TRUE value if the target begins with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of “myhostabc”: `http.req.url.hostname.startswith("myhost")`
`<text>.ENDSWITH(<string>)`	Returns a Boolean TRUE value if the target ends with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of “myhostabc”: `http.req.url.hostname.endswith("abc")`

Calculating the Length of a String

The <text>.LENGTH operation returns a numeric value that is equal to the number of characters (not bytes) in a string:

```
<text>.LENGTH
```

For example, you may want to identify request URLs that exceed a particular length. Following is an expression that implements this example:

```
HTTP.REQ.URL.LENGTH < 500
```

After taking a count of the characters or elements in a string, you can apply numeric operations to them. For more information, see Default Syntax Expressions: Working with Dates, Times, and Numbers.

Considering, Ignoring, and Changing Text Case

The following functions operate on the case (upper-case or lower-case) of the characters in the string.

Table 2. Functions for Considering, Ignoring, and Changing Text Case

Function	Description
`<text>.SET_TEXT_MODE(IGNORECASE|NOIGNORECASE)`	This function turns case sensitivity on or off for all text operations.
`<text>.TO_LOWER`	Converts the target to lowercase for a text block of up to 2 kilobyte (KB). Returns UNDEF if the target exceeds 2 KB. For example, the string “ABCd:” is converted to “abcd:”.
`<text>.TO_UPPER`	Converts the target to uppercase. Returns UNDEF if the target exceeds 2 KB. For example, the string “abcD:” is converted to “ABCD:”.

Stripping Specific Characters from a String

You can use the `STRIP_CHARS(<string>)` function to remove specific characters from the text that is returned by a default syntax expression prefix (the input string). All instances of the characters that you specify in the argument are stripped from the input string. You can use any text method on the resulting string, including the methods used for matching the string with a pattern set.

For example, in the expression `CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_"), the STRIP_CHARS(<string>)` function strips all periods (`.`), hyphens (`-`), and underscores (_) from the domain name returned by the prefix `CLIENT.UDP.DNS.DOMAIN`. If the domain name that is returned is "a.dom_ai_n-name", the function returns the string "adomainname".

In the following example, the resulting string is compared with a pattern set called "listofdomains":

```
CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")
```

Note: You cannot perform a rewrite on the string that is returned by the `STRIP_CHARS(<string>)` function.

The following functions strip matching characters from the beginning and end of a given string input.

Table 3. Functions for Stripping Characters From the Beginning or End of a String

Function	Description
`<text>.STRIP_START_CHARS(s)`	Strips matching characters from the beginning of the input string until the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is `TestLang` and `://_en_us:` is its value, `HTTP.RES.HEADER("TestLang").STRIP_START_CHARS(":/_")` strips the specified characters from the beginning of the value of the header until the first non-matching character `e` is found and returns `en_us:` as a string.
`<text>.STRIP_END_CHARS(s)`	Strips matching characters from the end of the input string to the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is `TestLang` and `://_en_us:` is its value, `HTTP.RES.HEADER("TestLang").STRIP_START_CHARS(":/_")` strips the specified characters from the end of the value of the header until the first non-matching character `s` is found and returns `://_en_us` as a string.

Complex Operations on Text

In addition to performing simple string matching, you can configure expressions that examine more complex aspects of text, including examining the length of a string and looking within a text block for patterns rather than specific strings.

Be aware of the following for any text-based operation:

- For any operation that takes a string argument, the string cannot exceed 255 characters.
- You can include white space when you specify a string in an expression.

Operations on the Length of a String

The following operations extract strings on the basis of a character count.

Table 1. String Operations Based on a Character Count

Character Count Operation	Description
`<text>.TRUNCATE(<count>)`	Returns a string after truncating the end of the target by the number of characters in <count>. If the entire string is shorter than <count>, nothing is returned.
`<text>.TRUNCATE(<character>, <count>)`	Returns a string after truncating the text after <character> by the number of characters specified in <count>.
`<text>.PREFIX(<character>, <count>)`	Selects the longest prefix in the target that has at most <count> occurrences of <character>.

	<text>.SUFFIX(<character>, <count>) Selects the longest suffix in the target that has at most <count> occurrences of <character>.
	
	For example, consider the following response body:
	
	`JLEwx`
	
	The following expression returns a value of "JLEwx":
	
	`http.res.body(100).suffix('L',1)`
	
	The following expression returns "LLEwx":
	
	`http.res.body(100).suffix('L',2)`

`<text>.SUBSTR(<starting_offset>, <length>)`	Select a string with <length> number of characters from the target object. Begin extracting the string after the <starting_offset>. If the number of characters after the offset are fewer than the value of the <length> argument, select all the remaining characters.
`<text>.SKIP(<character>, <count>)`	Select a string from the target after skipping over the longest prefix that has at most <count> occurrences of <character>.

Operations on a Portion of a String

You can extract a subset of a larger string by using one of the operations in the following table.

Table 2. Basic Operations on a Portion of a String

ext Operation	Description
>.BEFORE_STR(<string>)	Returns the text that precedes the first occurrence of <string>. If there is no match for <string>, the expression returns a text object of 0 length. Following is an example: `http.res.body(1024).after_str("start_string").before_str("end_string").conta`
>.AFTER_STR(<string>)	Returns the text that follows the first occurrence of <string>. If there is no match for <string>, the expression returns a text object of 0 length. Following is an example: `http.res.body(1024).after_str("start_string").before_str("end_string").conta`

>.BETWEEN(<starting g>, <ending string>)	Returns a Boolean TRUE value if the length of the text object is greater than or equal to the sum <sta <ending string> argument lengths, and if a prefix of the target matches <starting string>, and if the su matches <ending string>.
>.PREFIX(<prefix h>)	Returns the starting string from a target block of text that contains the number of characters in the < argument. If the <prefix length> argument exceeds the number of characters in the target, the entire string is se
>.SUFFIX(<suffix h>)	Returns the ending string from a target block of text that contains the number of characters in the <s argument. If the <suffix length> argument exceeds the number characters in the target, the entire str
>.SUBSTR(<string>)	Select the first block of text in the target that matches the <string>.
>.SKIP(<prefix length>)	Selects the text in the target after skipping over a <prefix length> number of characters. If the entire target has fewer characters than <prefix length>, the entire target is skipped.
>.STRIP_END_WS	Selects the text after removing white space from the end of the target.
>.STRIP_START_WS	Selects the text after removing white space from the beginning of the target.
>.UNQUOTE(<character>)	Selects the <character>, removes white space that immediately precedes and follows the <character> remaining text is quoted by <character>, this prefix also removes the quotes. For example, the operation UNQUOTE("") changes the following text: "abc xyz def " To the following: abc xyz def

Operations for Comparing the Alphanumeric Order of Two Strings

The COMPARE operation examines the first nonmatching character of two different strings. This operation is based on lexicographic order, which is the method used when ordering terms in dictionaries.

This operation returns the arithmetic difference between the ASCII values of the first nonmatching characters in the compared strings. The following differences are examples:

- The difference between “abc” and “abd” is -1 (based on the third pair-wise character comparison).
- The difference between “@” and “abc” is -33.
- The difference between “1” and “abc” is -47.

Following is the syntax for the COMPARE operation.

```
<text>.COMPARE(<string>)
```

Extracting an Integer from a String of Bytes That Represent Text

You can use the following functions to treat a string of bytes that represent text as a sequence of bytes, extract 8, 16, or 32 bits from the sequence, and then convert the extracted bits to an integer.

Table 3. Operations for Extracting an Integer from a String of Bytes That Represent Text

Function	Description
`<text>.GET_SIGNED8(<n>)`	Treats the string of bytes represented by text as a sequence of 8-bit signed integers and returns the integer at byte offset `n`. If the offset makes all or part of the value outside of the current text, an `UNDEF` condition is raised.
`<text>.GET_UNSIGNED8(<n>)`	Treats the string of bytes represented by text as a sequence of 8-bit unsigned integers and returns the integer at byte offset `n`. If the offset makes all or part of the value outside of the current text, an `UNDEF` condition is raised.

<text>.GET_SIGNED16(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 16 bits starting at byte offset n, and converts the extracted bit sequence to a 16-bit signed integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. **Note:** In NetScaler 9.2, the parameter n was an index into an array of 16-bit items. In NetScaler 9.3, the parameter is a byte offset. Therefore, if you used this function in NetScaler 9.2, after you upgrade to NetScaler 9.3, you must change n to 2*n to obtain the same results as you did earlier. For example, if the value of n before the upgrade was 4, you must change the value of n to 8. The parameter endianness also no longer takes the values that it did in NetScaler 9.2, which were 0 and 1. Instead, endianness accepts the mnemonic values mentioned earlier. **Example** HTTP.REQ.BODY(100).GET_SIGNED16(8, BIG_ENDIAN)

<text>.GET_UNSIGNED16(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 16 bits starting at byte offset n, and converts the extracted bit sequence to a 16-bit unsigned integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. **Note:** In NetScaler 9.2, the parameter n was an index into an array of 16-bit items. In NetScaler 9.3, the parameter is a byte offset. Therefore, if you used this function in NetScaler 9.2, after you upgrade to NetScaler 9.3, you must change n to 2*n to obtain the same results as you did earlier. For example, if the value of n before the upgrade was 4, you must change the value of n to 8. The parameter endianness also no longer takes the values that it did in NetScaler 9.2, which were 0 and 1. Instead, endianness accepts the mnemonic values mentioned earlier. **Example** HTTP.REQ.BODY(100).GET_UNSIGNED16(8, LITTLE_ENDIAN)

<text>.GET_SIGNED32(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 32 bits starting at byte offset n, and converts the extracted bit sequence to a 32-bit signed integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. **Note:** In NetScaler 9.2, the parameter n was an index into an array of 32-bit items. In NetScaler 9.3, the parameter is a byte offset. Therefore, if you used this function in NetScaler 9.2, after you upgrade to NetScaler 9.3, you must change n to 4*n to obtain the same results as you did earlier. For example, if the value of n before the upgrade was 4, you must change the value of n to 16. The parameter endianness also no longer takes the values that it did in NetScaler 9.2, which were 0 and 1. Instead, endianness accepts the mnemonic values mentioned earlier. **Example** HTTP.REQ.BODY(1000).GET_SIGNED32(12, BIG_ENDIAN)
<text>.GET_UNSIGNED32(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 32 bits starting at byte offset n, and returns the extracted bit sequence as part of a 64-bit unsigned long integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. **Example** HTTP.REQ.BODY(1000).GET_UNSIGNED32(30, LITTLE_ENDIAN)

Converting Text to a Hash Value

You can convert a text string to a hash value by using the .HASH function. This function returns a 31-bit positive integer as a result of the operation. Following is the format of the expression:

<text>.HASH

This function ignores case and white spaces. For example, after the operation, the two strings "Ab c" and "a bc" would produce the same hash value.

Encoding and Decoding Text by Applying the Base64 Encoding Algorithm

The following two functions encode and decode a text string by applying the Base64 encoding algorithm

Table 4. Functions for Encoding and Decoding a Text String by Using Base64 Encoding

Function	Description
`text.B64ENCODE`	Encodes the text string (designated by text) by applying the Base64 encoding algorithm.
`text.B64DECODE`	Decodes the Base64-encoded string (designated by text) by applying the Base64 decoding algorithm. The operation raises an `UNDEF` if text is not in B64-encoded format.

Refining the Search in a Rewrite Action by Using the EXTEND Function

The EXTEND function is used in rewrite actions that specify patterns or pattern sets and target the bodies of HTTP packets. When a pattern match is found, the EXTEND function extends the scope of the search by a predefined number of bytes on both sides of the matching string. A regular expression can then be used to perform a rewrite on matches in this extended region. Rewrite actions that are configured with the EXTEND function perform rewrites faster than rewrite actions that evaluate entire HTTP bodies using only regular expressions.

The format of the EXTEND function is EXTEND(*m*,*n*), where *m* and *n* are the number of bytes by which the scope of the search is extended before and after the matching pattern, respectively. When a match is found, the new search scope comprises m bytes that immediately precede the matching string, the string itself, and the n bytes that follow the string. A regular expression can then be used to perform a rewrite on a portion of this new string.

The EXTEND function can be used only if the rewrite action in which it is used fulfills the following requirements:

- The search is performed by using patterns or patterns sets (not regular expressions)
- The rewrite action evaluates only the bodies of HTTP packets.

Additionally, the EXTEND function can be used only with the following types of rewrite actions:

- replace_all
- insert_after_all
- delete_all
- insert_before_all

For example, you might want to delete all instances of "http://exampleurl.com/" and "http://exampleurl.au/" in the first 1000 bytes of the body. To do this, you can configure a rewrite action to search for all instances of the string "exampleurl," extend the scope of the search on both sides of the string when a match is found, and then use a regular expression to perform the rewrite in the extended region. The following example extends the scope of the search by 20 bytes to the left and 50 bytes to the right of the matching string:

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)'
-pattern exampleurl -refineSearch
'extend(20,50).regex_select(re#http://exampleurl.(com|au)#)'
```

Converting Text to Hexadecimal Format

The following function converts text to hexadecimal format and extracts the resulting string:

```
<text>.BLOB_TO_HEX(<string>)
```

For example, this function converts the byte string “abc” to “61:62:63”.

Encrypting and Decrypting Text

In default syntax expressions, you can use the ENCRYPT and DECRYPT functions to encrypt and decrypt text. Data encrypted by the ENCRYPT function on a given NetScaler appliance or high availability (HA) pair is intended for decryption by the DECRYPT function on the same NetScaler appliance or HA pair. The appliance supports the RC4, DES3, AES128, AES192, and AES256 encryption methods. The key value that is required for encryption is not user-specifiable. When an encryption method is set, the appliance automatically generates a random key value that is appropriate for the specified method. The default method is AES256 encryption, which is the most secure encryption method and the one that Citrix recommends.

You do not need to configure encryption unless you want to change the encryption method or you want the appliance to generate a new key value for the current encryption method.

Note: You can also encrypt and decrypt XML payloads. For information about the functions for encrypting and decrypting XML payloads, see Encrypting and Decrypting XML Payloads.

Configuring Encryption

During startup, the appliance runs the `set ns encryptionParams` command with, by default, the AES256 encryption method, and uses a randomly generated key value that is appropriate for AES256 encryption. The appliance also encrypts the key value and saves the command, with the encrypted key value, to the NetScaler configuration file. Consequently, the AES256 encryption method is enabled for the `ENCRYPT` and `DECRYPT` functions by default. The key value that is saved in the configuration file persists across reboots even though the appliance runs the command each time you restart it.

You can run the `set ns encryptionParams` command manually, or use the configuration utility, if you want to change the encryption method or if you want the appliance to generate a new key value for the current encryption method. To use the CLI to change the encryption method, set only the `method` parameter, as shown in "**Example 1: Changing the Encryption Method**." If you want the appliance to generate a new key value for the current encryption method, set the `method` parameter to the current encryption method and the `keyValue` parameter to an empty string (""), as shown in "**Example 2: Generating a New Key Value for the Current Encryption Method**." After you generate a new key value, you must save the configuration. If you do not save the configuration, the appliance uses the newly generated key value only until the next restart, after which it reverts to the key value in the saved configuration.

To configure encryption by using the NetScaler command line

At the NetScaler command prompt, type the following commands to configure encryption and verify the configuration:

- **`set ns encryptionParams -method`** `<method>` **`[-keyValue ""]`**
- **`show ns encryptionParams`**

Example 1. Changing the Encryption Method

For this example, assume that the current encryption method is AES256, and you want to change it to AES128.

```
> set ns encryptionParams -method AES128
Done
> show ns encryptionParams
      Method: AES128
Done
>
```

Example 2. Generating a New Key Value for the Current Encryption Method

In this example, the `show ns savedConfig | grep "set ns encryptionParams"` command displays the current encryption method and the encrypted key value in the saved configuration. The `encrypted` flag in the output of the command indicates that the key value is encrypted. The `set ns encryptionParams` command generates a new key

value for the current encryption method, and the `save ns config` command saves the configuration. The second `show ns savedConfig | grep "set ns encryptionParams"` verifies that a new key value has been generated.

```
> show ns savedConfig | grep "set ns encryptionParams"
set ns encryptionParams -method AES256 -keyValue ff0e316156e61a7adf9171e932aa90d02b54f75bd1c1663c4
> set ns encryptionParams -method AES256 -keyValue ""
 Done
> save ns config
 Done
> show ns savedConfig | grep "set ns encryptionParams"
set ns encryptionParams -method AES256 -keyValue fd0e316156e6172ddcab4eb8228f96900767917acae86d5b
>
```

Parameters for configuring encryption

method

The cipher method (and key length) used to encrypt and decrypt content. Possible values: NONE, RC4, DES3, AES128, AES192, AES256. Default: AES256.

keyValue

The base64-encoded key generation number, method, and key value. Omit this parameter if you enter a command to change the encryption method. To generate a new key value for the current encryption method, specify an empty string ("") as the value of this parameter. The parameter is passed implicitly, with its automatically generated value, to the NetScaler packet engines even when it is not included in the command. Passing the parameter to the packet engines enables the appliance to save the key value to the configuration file and to propagate the key value to the secondary appliance in a high availability setup.

To configure encryption by using the configuration utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the **Settings** area, click **Change Encryption parameters**.
3. In the **Change Encryption Parameters** dialog box, do one of the following:
   - To change the encryption method, in the **Method** list, select the encryption method that you want.
   - To generate a new key value for the current encryption method, click **Generate a new key for the selected method**.
4. Click **OK**.

Using the ENCRYPT and DECRYPT Functions in Expressions

You can use the ENCRYPT and DECRYPT functions with any expression prefix that returns text. For example, you can use the ENCRYPT and DECRYPT functions in rewrite policies for cookie encryption. In the following example, the rewrite actions encrypt a cookie named MyCookie, which is set by a back-end service, and decrypt the same cookie when it is returned by a client:

```
add rewrite action my-cookie-encrypt-action replace
"HTTP.RES.SET_COOKIE.COOKIE(\"MyCookie\").VALUE(0)"
"HTTP.RES.SET_COOKIE.COOKIE(\"MyCookie\").VALUE(0).ENCRYPT"
-bypassSafetyCheck YES

add rewrite action my-cookie-decrypt-action replace
"HTTP.REQ.COOKIE.VALUE(\"MyCookie\")"
"HTTP.REQ.COOKIE.VALUE(\"MyCookie\").DECRYPT" -bypassSafetyCheck YES
```

After you configure policies for encryption and decryption, save the configuration to bring the policies into effect.

Default Syntax Expressions: Working with Dates, Times, and Numbers

Most numeric data that the NetScaler appliance processes consists of dates and times. In addition to working with dates and times, the appliance processes other numeric data, such as the lengths of HTTP requests and responses. To process this data, you can configure default syntax expressions that process numbers.

A numeric expression consists of an expression prefix that returns a number and sometimes, but not always, an operator that can perform an operation on the number. Examples of expression prefixes that return numbers are `SYS.TIME.DAY`, `HTTP.REQ.CONTENT_LENGTH`, and `HTTP.RES.BODY.LENGTH`. Numeric operators can work with any prefix expression that returns data in numeric format. The GT(<int>) operator, for example, can be used with any prefix expression, such as `HTTP.REQ.CONTENT_LENGTH`, that returns an integer. Numeric expression prefixes and operators are also covered in Compound Operations for Numbers and Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data.

Format of Dates and Times in an Expression

When configuring a default syntax expression in a policy that works with dates and times (for example, the NetScaler system time or a date in an SSL certificate), you specify a time format as follows:

```
GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]
```

Where:

- **<yyyy>** is a four-digit year after GMT or LOCAL.
- **<month>** is a three-character abbreviation for the month, for example, Jan, Dec.
- **<d>** is a day of the week or an integer for the date.

  You cannot specify the day as Monday, Tuesday, and so on. You specify either an integer for a specific day of the month, or you specify a date as the first, second, third weekday of the month, and so on. Following are examples of specifying a day of the week:

  - Sun_1 is the first Sunday of the month.
  - Sun_3 is the third Sunday of the month.
  - Wed_3 is the third Wednesday of the month.
  - 30 is an example of an exact date in a month.
- **<h>** is the hour, for example, 10h.
- **<s>** is the number of seconds, for example, 30s.

The following example expression is true if the time is between10:00 a.m. and 5:30 p.m. local time, as determined from the time zone setting on the NetScaler. (Note that not all local time zones are supported.):

```
http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)
```

The following example expression is true for March and all months that follow March in the calendar year, based on GMT:

```
sys.time.ge(GMT 2008 Mar)
```

When you specify a date and time, note that the format is case sensitive and must preserve the exact number of blank spaces between entries.

Note: In an expression that requires two time values, both must use GMT or both must use LOCAL. You cannot mix the two in an expression.

Note: Unlike when you use the SYS.TIME prefix in a default syntax expression, if you specify SYS.TIME in a rewrite action, the NetScaler returns a string in conventional date format (for example, Sun, 06 Nov 1994 08:49:37 GMT). For example, the following rewrite action replaces the http.res.date header with the NetScaler system time in a conventional date format:

```
add rewrite action sync_date replace http.res.date sys.time
```

Expressions for the NetScaler System Time

The SYS.TIME expression prefix extracts the NetScaler system time. You can configure expressions that establish whether a particular event occurred at a particular time or within a particular time range according to the NetScaler system time.

The following table describes the expressions that you can create by using the SYS.TIME prefix.

Table 1. Expressions That Return NetScaler System Dates and Times

NetScaler Time Operation	Description
SYS.TIME.BETWEEN(<time1>, <time2>)	Returns a Boolean TRUE if the returned value is later than <time1> and earlier than <time2>. You format the <time1>, <time2> arguments as follows: • They must both be GMT or both LOCAL. • <time2> must be later than <time1>. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following: • sys.time.between(GMT 2004, GMT 2006) • sys.time.between(GMT 2004 Jan, GMT 2006 Nov) • sys.time.between(GMT 2004 Jan, GMT 2006) • sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) • sys.time.between(GMT 2005 May 1, GMT May 2005 1) • sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)
SYS.TIME.DAY	Returns the current day of the month as a number from 1 through 31.

SYS.TIME.EQ(<time>)	Returns a Boolean TRUE if the current time is equal to the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.eq(GMT 2005) (TRUE in this example.) • sys.time.eq(GMT 2005 Dec) (FALSE in this example.) • sys.time.eq(LOCAL 2005 May) (Evaluates to TRUE or FALSE in this example, depending on the current time zone.) • sys.time.eq(GMT 10h) (TRUE in this example.) • sys.time.eq(GMT 10h 30s) (TRUE in this example.) • sys.time.eq(GMT May 10h) (TRUE in this example.) • sys.time.eq(GMT Sun) (TRUE in this example.) • sys.time.eq(GMT May Sun_1) (TRUE in this example.)

SYS.TIME.GE(<time>)	Returns a Boolean TRUE if the current time is later than or equal to <time>. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.ge(GMT 2004) (TRUE in this example.) • sys.time.ge(GMT 2005 Jan) (TRUE in this example.) • sys.time.ge(LOCAL 2005 May) (TRUE or FALSE in this example, depending on the current time zone.) • sys.time.ge(GMT 8h) (TRUE in this example.) • sys.time.ge(GMT 30m) (FALSE in this example.) • sys.time.ge(GMT May 10h) (TRUE in this example.) • sys.time.ge(GMT May 10h 0m) (TRUE in this example.) • sys.time.ge(GMT Sun) (TRUE in this example.) • sys.time.ge(GMT May Sun_1) (TRUE in this example.)

SYS.TIME.GT(<time>)	Returns a Boolean TRUE if the time value is later than the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.gt(GMT 2004) (TRUE in this example.) • sys.time.gt(GMT 2005 Jan) (TRUE in this example.) • sys.time.gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • sys.time.gt(GMT 8h) (TRUE in this example.) • sys.time.gt(GMT 30m) (FALSE in this example.) • sys.time.gt(GMT May 10h) (FALSE in this example.) • sys.time.gt(GMT May 10h 0m) (TRUE in this example.) • sys.time.gt(GMT Sun) (FALSE in this example.) • sys.time.gt(GMT May Sun_1) (FALSE in this example.)
SYS.TIME.HOURS	Returns the current hour as an integer from 0 to 23.

SYS.TIME.LE(<time>)	Returns a Boolean TRUE if the current time value precedes or is equal to the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.le(GMT 2006) (TRUE in this example.) • sys.time.le(GMT 2005 Dec) (TRUE in this example.) • sys.time.le(LOCAL 2005 May) (TRUE or FALSE depending on the current timezone.) • sys.time.le(GMT 8h) (FALSE in this example.) • sys.time.le(GMT 30m) (TRUE in this example.) • sys.time.le(GMT May 10h) (TRUE in this example.) • sys.time.le(GMT Jun 11h) (TRUE in this example.) • sys.time.le(GMT Wed) (TRUE in this example.) • sys.time.le(GMT May Sun_1) (TRUE in this example.)
SYS.TIME.LT(<time>)	Returns a Boolean TRUE if the current time value precedes the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.lt(GMT 2006) (TRUE in this example.) • sys.time.lt.time.lt(GMT 2005 Dec) (TRUE in this example.) • sys.time.lt(LOCAL 2005 May) (TRUE or FALSE depending on the current time zone.) • sys.time.lt(GMT 8h) (FALSE in this example.) • sys.time.lt(GMT 30m) (TRUE in this example.) • sys.time.lt(GMT May 10h) (FALSE in this example.) • sys.time.lt(GMT Jun 11h) (TRUE in this example.) • sys.time.lt(GMT Wed) (TRUE in this example.) • sys.time.lt(GMT May Sun_1) (FALSE in this example.)

SYS.TIME.MINUTES	Returns the current minute as an integer from 0 to 59.
SYS.TIME.MONTH	Extracts the current month and returns an integer from 1 (January) to 12 (December).
SYS.TIME.RELATIVE_BOOT	Calculates the number of seconds to the closest previous or scheduled reboot, and returns an integer. If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.
SYS.TIME.RELATIVE_NOW	Calculates the number of seconds between the current NetScaler system time and the specified time, and returns an integer showing the difference. If the designated time is in the past, the integer is negative; if it is in the future, the integer is positive.
SYS.TIME.SECONDS	Extracts the seconds from the current NetScaler system time, and returns that value as an integer from 0 to 59.
SYS.TIME.WEEKDAY	Returns the current weekday as a value from 0 (Sunday) to 6 (Saturday).
SYS.TIME.WITHIN (<time1>, <time2>)	If you omit an element of time in <time1>, for example, the day or hour, it is assumed to have the lowest value in its range. If you omit an element in <time2>, it is assumed to have the highest value of its range. The ranges for the elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. If you specify the year, you must do so in both <time1> and <time2>. For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.within(GMT 2004, GMT 2006) (TRUE in this example.) • sys.time.within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.) • sys.time.within(GMT Feb, GMT) (TRUE, May is in the range of February to December.) • sys.time.within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.) • sys.time.within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE in this example.) • sys.time.within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone.)

SYS.TIME.YEAR	Extracts the year from the current system time and returns that value as a four-digit integer.

Expressions for SSL Certificate Dates

You can determine the validity period for SSL certificates by configuring an expression that contains the following prefix:

```
CLIENT.SSL.CLIENT_CERT
```

The following example expression matches a particular time for expiration with the information in the certificate:

```
client.ssl.client_cert.valid_not_after.eq(GMT 2009)
```

The following table describes time-based operations on SSL certificates. To obtain the expression you want, replace *certificate* in the expression in the the first column with the prefix expression, "`CLIENT.SSL.CLIENT_CERT`".

Table 1. Operations on Certificate (client.ssl.client_cert) Dates and Times

SSL Certificate Operation	Description
`<certificate>.VALID_NOT_AFTER`	Returns the last day before certificate expiration. The return fo is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).

`<certificate>.VALID_NOT_AFTER.BETWEEN(<time1>, <time2>)`	Returns a Boolean TRUE value if the certificate validity is betw the <time1> and <time2> arguments. Both <time1> and <time2> be fully specified. Following are examples: GMT 1995 Jan is fully specified. GMT Jan is not fully specified GMT 1995 20 is not fully specified. GMT Jan Mon_2 is not fully specified. The <time1> and <time2> arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>. For example, if it is GMT 2005 May 1 10h 15m 30s, and the first Sunday of the month, you can specify the following (evaluation results are in parentheses). • `. . .between(GMT 2004, GMT 2006) (TRUE)` • `. . .between(GMT 2004 Jan, GMT 2006 Nov) (TRU` • `. . .between(GMT 2004 Jan, GMT 2006) (TRUE)` • `. . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)` • `. . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)` • `. . .between(LOCAL 2005 May 1, LOCAL May 2005 (TRUE or FALSE, depending on the NetScaler sy time zone.)`
`<certificate>.VALID_NOT_AFTER.DAY`	Extracts the last day of the month that the certificate is valid, returns a number from 1 through 31, as appropriate for the dat

<certificate>.VALID_NOT_AFTER.EQ(<time>)	Returns a Boolean TRUE if the time is equal to the <time> argu For example, if the current time is GMT 2005 May 1 10h 15m 30 it is the first Sunday of the month, you can specify the followin (evaluation results for this example are in parentheses): • . . .eq(GMT 2005) (TRUE) • . . .eq(GMT 2005 Dec) (FALSE) • . . .eq(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone) • . . .eq(GMT 10h) (TRUE) • . . .eq(GMT 10h 30s) (TRUE) • . . .eq(GMT May 10h) (TRUE) • . . .eq(GMT Sun) (TRUE) • . . .eq(GMT May Sun_1) (TRUE)
<certificate>.VALID_NOT_AFTER.GE(<time>)	Returns a Boolean TRUE if the time value is greater than or equ the argument <time>. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • . . .ge(GMT 2004) (TRUE) • . . .ge(GMT 2005 Jan) (TRUE) • . . .ge(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.) • . . .ge(GMT 8h) (TRUE) • . . .ge(GMT 30m) (FALSE) • . . .ge(GMT May 10h) (TRUE) • . . .ge(GMT May 10h 0m) (TRUE) • . . .ge(GMT Sun) (TRUE) • . . .ge(GMT May Sun_1) (TRUE)

<certificate>.VALID_NOT_AFTER.GT(<time>)	Returns a Boolean TRUE if the time value is greater than the argument <time>. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • . . .gt(GMT 2004) (TRUE) • . . .gt(GMT 2005 Jan) (TRUE) • . . .gt(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.) • . . .gt(GMT 8h) (TRUE) • . . .gt(GMT 30m) (FALSE) • . . .gt(GMT May 10h) (FALSE) • . . .gt(GMT Sun) (FALSE) • . . .gt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_AFTER.HOURS	Extracts the last hour that the certificate is valid and returns th value as an integer from 0 to 23.
<certificate>.VALID_NOT_AFTER.LE(<time>)	Returns a Boolean TRUE if the time precedes or is equal to the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • . . .le(GMT 2006) (TRUE) • . . .le(GMT 2005 Dec) (TRUE) • . . .le(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.) • . . .le(GMT 8h) (FALSE) • . . .le(GMT 30m) (TRUE) • . . .le(GMT May 10h) (TRUE) • . . .le(GMT Jun 11h) (TRUE) • . . .le(GMT Wed) (TRUE) • . . .le(GMT May Sun_1) (TRUE)

<certificate>.VALID_NOT_AFTER.LT(<time>)	Returns a Boolean TRUE if the time precedes the <time> argum For example, if the current time is GMT 2005 May 1 10h 15m 30 it is the first Sunday of the month, you can specify the followin • . . .lt(GMT 2006) (TRUE) • . . .lt(GMT 2005 Dec) (TRUE) • . . .lt(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.) • . . .lt(GMT 8h) (FALSE) • . . .lt(GMT 30m) (TRUE) • . . .lt(GMT May 10h) (FALSE) • . . .lt(GMT Jun 11h) (TRUE) • . . .lt(GMT Wed) (TRUE) • . . .lt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_AFTER.MINUTES	Extracts the last minute that the certificate is valid and returns value as an integer from 0 to 59.
<certificate>.VALID_NOT_AFTER.MONTH	Extracts the last month that the certificate is valid and returns value as an integer from 1 (January) to 12 (December).
<certificate>.VALID_NOT_AFTER.RELATIVE_BOOT	Calculates the number of seconds to the closest previous or scheduled reboot and returns an integer. If the closest boot tim the past, the integer is negative. If it is in the future, the integ positive.
<certificate>.VALID_NOT_AFTER.RELATIVE_NOW	Calculates the number of seconds between the current system and the specified time and returns an integer. If the time is in past, the integer is negative; if it is in the future, the integer is positive.
<certificate>.VALID_NOT_AFTER.SECONDS	Extracts the last second that the certificate is valid and returns value as an integer from 0 to 59.
<certificate>.VALID_NOT_AFTER.WEEKDAY	Extracts the last weekday that the certificate is valid. Returns number between 0 (Sunday) and 6 (Saturday) to give the weekd the time value.

<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>)	Returns a Boolean TRUE if the time lies within all the ranges de by the elements in <time1> and <time2>. If you omit an element of time from <time1>, it is assumed to h the lowest value in its range. If you omit an element from <tim is assumed to have the highest value of its range. If you specify year in <time1>, you must specify it in <time2>. The ranges for elements of time are as follows: month 1-12, da 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. F the result to be TRUE, each element in the time must exist in t corresponding range that you specify in <time1>, <time2>. For example, if time is GMT 2005 May 10 10h 15m 30s, and it is second Tuesday of the month, you can specify the following (evaluation results are in parentheses): • . . .within(GMT 2004, GMT 2006) (TRUE) • . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALS May is not in the range of January to March.) • . . .within(GMT Feb, GMT) (TRUE, May is in th range for February to December) • . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday lies within the range of the f Sunday through the third Sunday) • . . .within(GMT 2005 May 1 10h, GMT May 2005 17h) (TRUE) • . . .within(LOCAL 2005 May 1, LOCAL May 2005 (TRUE or FALSE, depending on the NetScaler sy time zone)
<certificate>.VALID_NOT_AFTER.YEAR	Extracts the last year that the certificate is valid and returns a four-digit integer.
<certificate>.VALID_NOT_BEFORE	Returns the date that the client certificate becomes valid. The return format is the number of seconds since GMT January 1970 (0 hours, 0 minutes, 0 seconds).

`<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>)`	Returns a Boolean TRUE if the time value is between the the tw time arguments. Both <time1> and <time2> arguments must be specified. Following are examples: • GMT 1995 Jan is fully specified. • GMT Jan is not fully specified. • GMT 1995 20 is not fully specified. • GMT Jan Mon_2 is not fully specified. The time arguments must be both GMT or both LOCAL, and <tim must be greater than <time1>. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • `. . .between(GMT 2004, GMT 2006) (TRUE)` • `. . .between(GMT 2004 Jan, GMT 2006 Nov) (TRU` • `. . .between(GMT 2004 Jan, GMT 2006) (TRUE)` • `. . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)` • `. . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)` • `. . .between(LOCAL 2005 May 1, LOCAL May 2005 (TRUE or FALSE, depending on the NetScaler sy time zone.)`
`<certificate>.VALID_NOT_BEFORE.DAY`	Extracts the last day of the month that the certificate is valid a returns that value as a number from 1 through 31 representing day.

`<certificate>.VALID_NOT_BEFORE.EQ(<time>)`	Returns a Boolean TRUE if the time is equal to the <time> argu For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • `. . .eq(GMT 2005) (TRUE)` • `. . .eq(GMT 2005 Dec) (FALSE)` • `. . .eq(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.)` • `. . .eq(GMT 10h) (TRUE)` • `. . .eq(GMT 10h 30s) (TRUE)` • `. . .eq(GMT May 10h) (TRUE)` • `. . .eq(GMT Sun) (TRUE)` • `. . .eq(GMT May Sun_1) (TRUE)`
`<certificate>.VALID_NOT_BEFORE.GE(<time>)`	Returns a Boolean TRUE if the time is greater than (after) or eq the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results are in parentheses): • `. . .ge(GMT 2004) (TRUE)` • `. . .ge(GMT 2005 Jan) (TRUE)` • `. . .ge(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.)` • `. . .ge(GMT 8h) (TRUE)` • `. . .ge(GMT 30m) (FALSE)` • `. . .ge(GMT May 10h) (TRUE)` • `. . .ge(GMT May 10h 0m) (TRUE)` • `. . .ge(GMT Sun) (TRUE)` • `. . .ge(GMT May Sun_1) (TRUE)`

<certificate>.VALID_NOT_BEFORE.GT(<time>)	Returns a Boolean TRUE if the time occurs after the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results are in parentheses): • . . .gt(GMT 2004) (TRUE) • . . .gt(GMT 2005 Jan) (TRUE) • . . .gt(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.) • . . .gt(GMT 8h) (TRUE) • . . .gt(GMT 30m) (FALSE) • . . .gt(GMT May 10h) (FALSE) • . . .gt(GMT May 10h 0m) (TRUE) • . . .gt(GMT Sun) (FALSE) • . . .gt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_BEFORE.HOURS	Extracts the last hour that the certificate is valid and returns th value as an integer from 0 to 23.
<certificate>.VALID_NOT_BEFORE.LE(<time>)	Returns a Boolean TRUE if the time precedes or is equal to the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • . . .le(GMT 2006) (TRUE) • . . .le(GMT 2005 Dec) (TRUE) • . . .le(LOCAL 2005 May) (TRUE or FALSE,depend on the current time zone.) • . . .le(GMT 8h) (FALSE) • . . .le(GMT 30m) (TRUE) • . . .le(GMT May 10h) (TRUE) • . . .le(GMT Jun 11h) (TRUE) • . . .le(GMT Wed) (TRUE) • . . .le(GMT May Sun_1) (TRUE)

<certificate>.VALID_NOT_BEFORE.LT(<time>)	Returns a Boolean TRUE if the time precedes the <time> argum For example, if the time value is GMT 2005 May 1 10h 15m 30s, is the first Sunday of the month of May in 2005, you can specify following (evaluation results for this example are in parenthese • . . .lt(GMT 2006) (TRUE) • . . .lt(GMT 2005 Dec) (TRUE) • . . .lt(LOCAL 2005 May) (TRUE or FALSE, deper on the current time zone.) • . . .lt(GMT 8h) (FALSE) • . . .lt(GMT 30m) (TRUE) • . . .lt(GMT May 10h) (FALSE) • . . .lt(GMT Jun 11h) (TRUE) • . . .lt(GMT Wed) (TRUE) • . . .lt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_BEFORE.MINUTES	Extracts the last minute that the certificate is valid. Returns th current minute as an integer from 0 to 59.
<certificate>.VALID_NOT_BEFORE.MONTH	Extracts the last month that the certificate is valid. Returns the current month as an integer from 1 (January) to 12 (December)
<certificate>.VALID_NOT_BEFORE.RELATIVE_BOOT	Calculates the number of seconds to the closest previous or scheduled NetScaler reboot and returns an integer. If the closes boot time is in the past, the integer is negative; if it is in the fu the integer is positive.
<certificate>.VALID_NOT_BEFORE.RELATIVE_NOW	Returns the number of seconds between the current NetScaler time and the specified time as an integer. If the designated tim the past, the integer is negative. If it is in the future, the integ positive.
<certificate>.VALID_NOT_BEFORE.SECONDS	Extracts the last second that the certificate is valid. Returns th current second as an integer from 0 to 59.
<certificate>.VALID_NOT_BEFORE.WEEKDAY	Extracts the last weekday that the certificate is valid. Returns weekday as a number between 0 (Sunday) and 6 (Saturday).

<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>, <time2>)	Returns a Boolean TRUE if each element of time exists within th range defined in the <time1>, <time2> arguments. If you omit an element of time from <time1>, it is assumed to h the lowest value in its range. If you omit an element of time fro <time2>, it is assumed to have the highest value in its range. If specify a year in <time1>, it must be specified in <time2>. The ranges for elements of time are as follows: month 1-12, day 1-3 weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. For example, if the time is GMT 2005 May 10 10h 15m 30s, and the second Tuesday of the month, you can specify the following (evaluation results are in parentheses): • . . .within(GMT 2004, GMT 2006) (TRUE) • . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE is not in the range of January to March.) • . . .within(GMT Feb, GMT) (TRUE, May is in the rang February to December.) • . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the sec Tuesday is between the first Sunday and the third Sunday.) • . . .within(GMT 2005 May 1 10h, GMT May 2005 17h) (TRUE) • . . .within(LOCAL 2005 May 1, LOCAL May 2005 (TRUE or FALSE, depending on the NetScaler system time zo
<certificate>.VALID_NOT_BEFORE.YEAR	Extracts the last year that the certificate is valid. Returns the current year as a four-digit integer.

Expressions for HTTP Request and Response Dates

The following expression prefixes return the contents of the HTTP Date header as text or as a date object. These values can be evaluated as follows:

- **As a number.** The numeric value of an HTTP Date header is returned in the form of the number of seconds since Jan 1 1970.

  For example, the expression `http.req.date.mod(86400)` returns the number of seconds since the beginning of the day. These values can be evaluated using the same operations as other non-date-related numeric data. For more information, see Expression Prefixes for Numeric Data Other Than Date and Time.

- **As an HTTP header.** Date headers can be evaluated using the same operations as other HTTP headers.

  For more information, see Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data.

- **As text.** Date headers can be evaluated using the same operations as other strings.

For more information, see Default Syntax Expressions: Evaluating Text.

Table 1. Prefixes That Evaluate HTTP Date Headers

Prefix	Description
`HTTP.REQ.DATE`	Returns the contents of the HTTP Date header as text or as a date object.The date formats recognized are: RFC822. Sun, 06 Jan 1980 08:49:37 GMT RFC850. Sunday, 06-Jan-80 09:49:37 GMT ASCTIME. Sun Jan 6 08:49:37 1980
`HTTP.RES.DATE`	Returns the contents of the HTTP Date header as text or as a date object.The date formats recognized are: RFC822. Sun, 06 Jan 1980 8:49:37 GMT RFC850. Sunday, 06-Jan-80 9:49:37 GMT ASCTIME. Sun Jan 6 08:49:37 1980

Generating the Day of the Week, as a String, in Short and Long Formats

The functions, WEEKDAY_STRING_SHORT and WEEKDAY_STRING, generate the day of the week, as a string, in short and long formats, respectively. The strings that are returned are always in English. The prefix used with these functions must return the day of the week in integer format and the acceptable range for the value returned by the prefix is 0-6. Therefore, you can use any prefix that returns an integer in the acceptable range. An UNDEF condition is raised if the returned value is not in this range or if memory allocation fails.

Following are the descriptions of the functions:

Table 1. Functions That Generate the Day of the Week, as a String, in Short and Long Formats

Function	Description
<prefix>.WEEKDAY_STRING_SHORT	Returns the day of the week in short format. The short form is always 3 characters long with an initial capital and the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT returns Sun if the value returned by the WEEKDAY function is 0 and Sat if the value returned by the prefix is 6.
<prefix>.WEEKDAY_STRING	Returns the day of the week in long format. The long form always has an initial capital, with the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING returns Sunday if the value returned by the WEEKDAY function is 0 and Saturday if the value returned by the prefix is 6.

Expression Prefixes for Numeric Data Other Than Date and Time

In addition to configuring expressions that operate on time, you can configure expressions for the following types of numeric data:

- The length of HTTP requests, the number of HTTP headers in a request, and so on.

  For more information, see Expressions for Numeric HTTP Payload Data Other Than Dates.

- IP and MAC addresses.

  For more information, see Expressions for IP Addresses and IP Subnets.

- Client and server data in regard to interface IDs and transaction throughput rate.

  For more information, see Expressions for Numeric Client and Server Data.

- Numeric data in client certificates other than dates.

  For information on these prefixes, including the number of days until certificate expiration and the encryption key size, see Prefixes for Numeric Data in SSL Certificates.

Converting Numbers to Text

The following functions produce binary strings from a number returned by an expression prefix. These functions are particularly useful in the TCP rewrite feature as replacement strings for binary data. For more information about the TCP rewrite feature, see Rewrite.

All the functions return a value of type `text`. The endianness that some of the functions accept as a parameter is either LITTLE_ENDIAN or BIG_ENDIAN.

Table 1. Functions That Produce a Binary String From a Number

	Description
`TRING`	Produces an 8-bit signed binary string representing the number. If the value is out raised. **Example** `HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STRING`
`_STRING`	Produces an 8-bit unsigned binary string representing the number. If the value is ou raised. **Example** `HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED8_STRING`
`STRING(<endianness>)`	Produces a 16-bit signed binary string representing the number. If the value is out raised. **Example** `HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)`
`6_STRING(<endianness>)`	Produces a 16-bit unsigned binary string representing the number. If the value is ou raised. **Example** `HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LITTLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)`
`STRING(<endianness>)`	Produces a 32-bit signed binary string representing the number. **Example** `HTTP.REQ.BODY(100).AFTER_STR("delim").GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)`

ber>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is ou raised. **Example** `HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIGNED_LONG_AT`
ber>.UNSIGNED16_STRING(<endianness>)	Produces a 16-bit unsigned binary string representing the number. If the value is ou raised. **Example** `HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LITTLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UNSIGNED16_ST`
ber>.UNSIGNED32_STRING(<endianness>)	Produces a 32-bit unsigned binary string representing the number. If the value is ou raised. **Example** `HTTP.REQ.BODY(100).AFTER_STR("delim2").GET_UNSIGNED32(0, BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)`

Virtual Server Based Expressions

The SYS.VSERVER("<vserver-name>") expression prefix enables you to identify a virtual server. You can use the following functions with this prefix to retrieve information related to the specified virtual server:

- **THROUGHPUT.** Returns the throughput of the virtual server in Mbps (Megabits per second). The value returned is an unsigned long number.

  Usage: SYS.VSERVER("vserver").THROUGHPUT

- **CONNECTIONS.** Returns the number of connections being managed by the virtual server. The value returned is an unsigned long number.

  Usage: SYS.VSERVER("vserver").CONNECTIONS

- **STATE.** Returns the state of the virtual server. The value returned is UP, DOWN, or OUT_OF_SERVICE. One of these values can therefore be passed as an argument to the EQ() operator to perform a comparison that results in a Boolean TRUE or FALSE.

  Usage: SYS.VSERVER("vserver").STATE

- **HEALTH.** Returns the percentage of services in an UP state for the specified virtual server. The value returned is an integer.

  Usage: SYS.VSERVER("vserver").HEALTH

- **RESPTIME.** Returns the response time as a number. The unit is millisecond. Response time is the average TTFB (Time To First Byte) from all the services bound to the virtual server. The value returned is an integer.

  Usage: SYS.VSERVER("vserver").RESPTIME

- **SURGECOUNT.** Returns the number of requests in the surge queue of the virtual server. The value returned is an integer.

  Usage: SYS.VSERVER("vserver").SURGECOUNT

Example 1

The following rewrite policy aborts rewrite processing if the number of connections at the load balancing virtual server LBvserver exceeds 10000:

```
add rewrite policy norewrite_pol
sys.vserver("LBvserver").connections.gt(10000) norewrite
```

Example 2

The following rewrite action inserts a custom header, TP, whose value is the throughout at the virtual server LBvserver:

```
add rewrite action tp_header insert_http_header TP
SYS.VSERVER("LBvserver").THROUGHPUT
```

Example 3

The following audit log message action writes the average TTFB of the services bound to a virtual server, to the newnslog log file:

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS
Response Time to Servers:\" + sys.vserver(\"ssllb\").resptime + \"
millisec\"" -logtoNewnslog YES -bypassSafetyCheck YES
```

Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data

You can configure default syntax expressions to evaluate and process the payload in HTTP requests and responses. The payload associated with an HTTP connection includes the various HTTP headers (both standard and custom headers), the body, and other connection information such as the URL. Additionally, you can evaluate and process the payload in a TCP or UDP packet. For HTTP connections, for example, you can check whether a particular HTTP header is present or if the URL includes a particular query parameter.

You can configure expressions to transform the URL encoding and apply HTML or XML “safe” coding for subsequent evaluation. You can also use XPATH and JSON prefixes to evaluate date in XML and JSON files, respectively.

You can also use text-based and numeric default syntax expressions to evaluate HTTP request and response data. For more information, see Default Syntax Expressions: Evaluating Text and Default Syntax Expressions: Working with Dates, Times, and Numbers.

About Evaluating HTTP and TCP Payload

The payload of an HTTP request or response consists of HTTP protocol information such as headers, a URL, body content, and version and status information. When you configure a default syntax expression to evaluate HTTP payload, you use a default syntax expression prefix and, if necessary, an operator.

For example, you use the following expression, which includes the `http.req.header(“<header_name>“)` prefix and the `exists` operator, if you want to determine whether an HTTP connection includes a custom header named “myHeader”:

```
http.req.header("myHeader").exists
```

You can also combine multiple default syntax expressions with Boolean and arithmetic operators. For example, the following compound expression could be useful with various NetScaler features, such as Integrated Caching, Rewrite, and Responder. This expression first uses the && Boolean operator to determine whether an HTTP connection includes the Content-Type header with a value of "text/html." If that operation returns a value of FALSE, the expression determines whether the HTTP connection includes a "Transfer-Encoding" or "Content-Length" header.

```
(http.req.header("Content-Type").exists &&
http.req.header("Content-Type").eq("text/html")) ||
(http.req.header("Transfer-Encoding").exists) ||
(http.req.header("Content-Length").exists)
```

The payload of a TCP or UDP packet is the data portion of the packet. You can configure default syntax expressions to examine features of a TCP or UDP packet, including the following:

- Source and destination domains
- Source and destination ports
- The text in the payload
- Record types

The following expression prefixes extract text from the body of the payload:

- **HTTP.REQ.BODY(integer).** Returns the body of an HTTP request as a multiline text object, up to the character position designated in the *integer* argument. If there are fewer characters in the body than is specified in the argument, the entire body is returned.
- **HTTP.RES.BODY(integer).** Returns a portion of the HTTP response body. The length of the returned text is equal to the number in the *integer* argument. If there are fewer characters in the body than is specified in integer, the entire body is returned.
- **CLIENT.TCP.PAYLOAD(integer).** Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the

integer argument.

Following is an example that evaluates to TRUE if a response body of 1024 bytes contains the string “https”, and this string occurs after the string “start string” and before the string “end string”:

```
http.res.body(1024).after_str("start_string").before_str("end_string").contains("h
```

Note: You can apply any text operation to the payload body. For information on operations that you can apply to text, see Default Syntax Expressions: Evaluating Text.

Expressions for HTTP and Cache-Control Headers

One common method of evaluating HTTP traffic is to examine the headers in a request or a response. A header can perform a number of functions, including the following:

- Provide cookies that contain data about the sender.
- Identify the type of data that is being transmitted.
- Identify the route that the data has traveled (the Via header).

Note: Note that if an operation is used to evaluate both header and text data, the header-based operation always overrides the text-based operation. For example, the AFTER_STR operation, when applied to a header, overrides text-based AFTER_STR operations for all instances of the current header type.

Prefixes for HTTP Headers

The following table describes expression prefixes that extract HTTP headers.

Table 1. Prefixes That Extract HTTP Headers

	Description
eader_name>")	Returns the contents of the HTTP header specified by the <header_nam cannot exceed 32 characters. Note that this prefix returns the value from the Host header by default you need to typecast it as follows: `http.req.header("host").typecast_http_hostname_t` For more information on typecasting, see Typecasting Data.
R	Returns the contents of the complete set of HTTP header fields includi "GET /brochures/index.html HTTP/1.1") and the terminating \r\n\r\
	Returns the contents of the HTTP Date header.The following date form RFC822. Sun, 06 Jan 1980 8:49:37 GMT RFC850. Sunday, 06-Jan-80 9:49:37 GMT ASCII TIME. Sun Jan 6 08:49:37 1980 To evaluate a Date header as a date object, see Default Syntax Express and Numbers.

	(Name/Value List) Returns the contents of the HTTP Cookie header.
	Returns the HTTP transaction ID. The value is a function of an internal time and system MAC address.
`eader_name>")`	Returns the contents of the HTTP header specified by the <header_nam cannot exceed 32 characters.
`R`	Returns the contents of the complete set of HTTP header fields includi "HTTP/1.1 200 OK") and the terminating `\r\n\r\n` sequence.
`2`	Returns the HTTP Set-Cookie header object in a response.

`("<name>")`
 	
`2("<name>")`
	Returns the cookie of the specified name if it is present. If it is not pre 0. Returns UNDEF if more than 15 Set-Cookie headers are present and t these headers.

`("<name>").DOMAIN`
 	
`2("<name>").DOMAIN`
	Returns the value of the first Domain field in the cookie. For example, Customer = "ABC"; DOMAIN=".abc.com"; DOMAIN=.xyz.com, the followin
	
	`http.res.set_cookie.cookie("customer").domain`
	
	A string of zero length is returned if the Domain field or its value is abs

`.EXISTS("<name>")`
 	
`2.EXISTS("<name>")`
	Returns a Boolean TRUE if a Cookie with the name specified in the <nar Set-Cookie header.
	
	This prefix returns UNDEF if more than 15 Set-Cookie headers are prese the first 15 headers.

`.COOKIE("<name>").EXPIRES`
 	
`2.COOKIE("<name>").EXPIRES`
	Returns the Expires field of the cookie. This is a date string that can be object, or as text. If multiple Expires fields are present, the first one is absent, a text object of length zero is returned.
	
	To evaluate the returned value as a time object, see Default Syntax Ex Times, and Numbers.

`.COOKIE("<name>").PATH|PATH.GET(n)`
 	
`2.COOKIE("<name>").PATH|PATH.GET(n)`
	Returns the value of Path field of the cookie as a slash- (“/”) separated are treated as single slash. If multiple Path fields are present, the valu
	
	For example, the following is a cookie with two path fields:
	
	`Set-Cookie : Customer = "ABC"; PATH="/a//b/c"; PATH=`
	
	The following expression returns `/a//b/c` from this cookie:
	
	`http.res.set_cookie.cookie("Customer").path`
	
	The following expression returns b:
	
	`http.res.set_cookie.cookie("Customer").path.get(2)`
	
	Quotes are stripped from the returned value. A string of zero length is value is absent.

.COOKIE("<name>").PATH.IGNORE_EMPTY_ELEMENTS
 	
2.COOKIE("<name>").PATH.IGNORE_EMPTY_ELEMENTS
	Ignores the empty elements in the list. For example, in the list a=10,b= the list is , and the list has an empty element following a=10. The elem an empty element.
	
	As another example, in the following expression, if a request contains following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_list_t(',').i`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_list_t(',').c`

.COOKIE("<name>").PORT 2.COOKIE("<name>").PORT	Returns the value of Port field of the cookie. Operate as a comma-sepa For example, the following expression returns 80. 2580 from Set-Cookie PATH="/a/b/c"; PORT= "80, 2580": `http.res.set_cookie.cookie("ABC").port` A string of zero length is returned if the Port field or value is absent.

.COOKIE("<name>").PORT.IGNORE_EMPTY_ELEMENTS
 	
2.COOKIE("<name>").PORT.IGNORE_EMPTY_ELEMENTS
	Ignores the empty elements in the list. For example, in the list a=10,b= the list is , and the list has an empty element following a=10. The elem an empty element.
	
	As another example, in the following expression, if a request contains following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_list_t(',').i`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_list_t(',').c`

.COOKIE("<name>").VERSION 2.COOKIE("<name>").VERSION	Returns the value of the first Version field in the cookie as a decimal in For example, the following expression returns 1 from the cookie Set-Co "1"; VERSION = "0" `http.res.set_cookie.cookie("CUSTOMER").version` A zero is returned if the Version field or its value is absent or if the val
.COOKIE("<name>", <integer>) 2.COOKIE("<name>", <integer>)	Returns the nth instance (0-based) of the cookie with the specified nam text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the
.COOKIE("<name>", <integer>).DOMAIN 2.COOKIE("<name>", <integer>).DOMAIN	Returns the value of the Domain field of the first cookie with the speci following expression returns a value of abc.com from the cookie Set-Co DOMAIN=".abc.com"; DOMAIN=.xyz.com `http.res.set_cookie.cookie("CUSTOMER").domain` A string of zero length is returned if the Domain field or its value is abs

.COOKIE("<name>", <integer>).EXPIRES 2.COOKIE("<name>", <integer>).EXPIRES	Returns the *n*th instance (0-based) of the Expires field of the cookie wi string. The value can be operated upon as a time object that supports Expires attribute is absent a string of length zero is returned.

.COOKIE("<name>", <integer>).PATH | PATH.GET(i)
 	
2.COOKIE("<name>", <integer>).PATH |
	Returns the value of the Path field of the nth cookie, as a '/' separated
	single /.
	
	For example, the following expression returns /a//b/c from the cookie
	PATH="/a//b/c"; PATH= "/x/y/z"
	
	`http.res.set_cookie.cookie("CUSTOMER").path`
	
	The following returns b:
	
	`http.res.set_cookie.cookie("CUSTOMER").path.get(2)`
	
	A string of zero length is returned if the Path field or its value is absen

.COOKIE("<name>",
ORE_EMPTY_ELEMENTS
 	
2.COOKIE("<name>",
ORE_EMPTY_ELEMENTS
	Ignores the empty elements in the list. For example, in the list a=10,b=
	the list is , and the list has an empty element following a=10. The elem
	an empty element.
	
	As another example, in the following expression, if a request contains
	following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_list_t(',').i`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_list_t(',').c`

.COOKIE("<name>", <integer>).PORT 2.COOKIE("<name>", <integer>).PORT	Returns the value or values of the Port field of the named cookie as a ' following expression returns 80, 2580 from the cookie Set-Cookie : Cust PORT= "80, 2580" `http.res.set_cookie.cookie("ABC").port` A string of zero length is returned if the Port field or its value is absen

.COOKIE("<name>",
ORE_EMPTY_ELEMENTS
 	
2.COOKIE("<name>",
ORE_EMPTY_ELEMENTS
	Ignores the empty elements in the list. For example, in the list a=10,b=
	the list is , and the list has an empty element following a=10. The elem
	an empty element.
	
	As another example, in the following expression, if a request contains
	following expression returns a value of 4:
	
	`http.req.header("Cust_Header").typecast_list_t(',').i`
	
	The following expression returns a value of 5:
	
	`http.req.header("Cust_Header").typecast_list_t(',').c`

.COOKIE("<name>", <integer>).VERSION 2.COOKIE("<name>", <integer>).VERSION	Returns the value of Version field of the *n*th cookie as a decimal intege A string of zero length is returned if the Port field or its value is absent
	Returns the HTTP transaction ID. The value is a function of an internal time and system MAC address.

Operations for HTTP Headers

The following table describes operations that you can specify with the prefixes for HTTP headers.

Table 2. Operations That Evaluate HTTP Headers

HTTP Header Operation	Description
`http header .EXISTS`	Returns a Boolean TRUE if an instance of the specified header type exists. Following is an example: `http.req.header("Cache-Control").exists`

`http header.CONTAINS" http header . CONTAINS("<string>")`	Returns a Boolean TRUE if the <string> argument appears in any instance of the header value.
	
	Note: This operation overrides any text-based Contains operations on all instances of the current header type.
	
	Following is an example of request with two headers:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: abc\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: def\r\n`
	
	`\r\n`
	
	The following returns a Boolean TRUE:
	
	`http.res.header("MyHeader"). contains("de")`
	
	The following returns FALSE. Note that the NetScaler does not concatenate the different values.
	
	`http.res.header("MyHeader"). contains("bcd")`

`http header .COUNT`	Returns the number of headers in a request or response, to a maximum of 15 headers of the same type. The result is undefined if there are more than 15 instances of the header.
	
	Following is sample data in a request:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: abc\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: def\r\n`
	
	`\r\n`
	
	When evaluating the preceding request, the following returns a count of 2:
	
	`http.res.header("MyHeader").count`

`http header.AFTER_STR("<string>")`	Extracts the text that follows the first occurrence of the <string> argument.The headers are evaluated from the last instance to the first.
	
	Following is an example of a request:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: 111abc\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: 111def\r\n`
	
	`\r\n`
	
	The following extracts the string "def" from the last instance of MyHeader. This is value "111def."
	
	`http.res.header("MyHeader").after_str("111")`
	
	The following extracts the string "c" from the first instance of MyHeader. This is the value "abc111."
	
	`http.res.header("MyHeader").after_str("1ab")`

`http header.BEFORE_STR("<string>")`	Extracts the text that appears prior to the first occurrence of the input <string> argument.The headers are evaluated from the last instance to the first.
	
	Following is an example of a request that contains headers:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: abc111\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: def111\r\n`
	
	`\r\n`
	
	The following extracts the string "def" from the last instance of MyHeader. This is the value "def111."
	
	`http.res.header("MyHeader").before_str("111")`
	
	The following extracts the string "a" from the first instance of MyHeader. This is the value "abc111."
	
	`http.res.header("MyHeader").before_str("bc1")`

`http header.INSTANCE(<instance number>)`	An HTTP header can occur multiple times in a request or a response. This operation returns the header that occurs <instance number> of places before the final instance. For example, instance(0) selects the last instance of the current type, instance(1) selects the next-to-last instance, and so on. This prefix cannot be used in bidirectional policies.
	
	The <instance number> argument cannot exceed 14. Following is an example of a request with two headers:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: abc\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: def\r\n`
	
	`\r\n`
	
	The following returns a text object that refers to "MyHeader: abc\r\n":
	
	`http.res.header("MyHeader").instance(1)`

`http header.SUBSTR("<string>")`	Extracts the text that matches the <string> argument. The headers are evaluated from the last instance to the first. Following is an example of a request with two headers that contain the string “111”:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: abc111\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: 111def\r\n`
	
	`\r\n`
	
	The following returns "111" from the last instance of MyHeader. This is the header with the value "111def."
	
	`http.res.header("MyHeader").substr("111")`

`http header.VALUE(<instance number>)`	An HTTP header can occur multiple times in a request or a response. VALUE(0) selects the value in the last instance, VALUE(1) selects the value in the next-to-last instance, and so on. The <instance number> argument cannot exceed 14.
	
	Following is an example of a request with two headers:
	
	`HTTP/1.1 200 OK\r\n`
	
	`MyHeader: abc\r\n`
	
	`Content-Length: 200\r\n`
	
	`MyHeader: def\r\n`
	
	`\r\n`
	
	The following returns "abc\r\n":
	
	`http.res.header("MyHeader").value(1)`

Prefixes for Cache-Control Headers

The following prefixes apply specifically to Cache-Control headers.

Table 3. Prefixes That Extract Cache-Control Headers

HTTP Header Prefix	Description
`HTTP.REQ.CACHE_CONTROL`	Returns a Cache-Control header in an HTTP request.
`HTTP.RES.CACHE_CONTROL`	Returns a Cache-Control header in an HTTP response.

Operations for Cache-Control Headers

You can apply any of the operations for HTTP headers to Cache-Control headers. For more information, see Operations for HTTP Headers.

In addition, the following operations identify specific types of Cache-Control headers. See RFC 2616 for information about these header types.

Table 4. Operations That Evaluate Cache-Control Headers

HTTP Header Operation	Description
`Cache-Control header.NAME(<integer>)`	Returns as a text value the name of the Cache-Control header that corresponds to the nth component in a name-value list, as specified by <integer>. The index of the name-value component is 0-based. If the <integer> that is specified by the integer argument is greater than the number of components in the list, a zero-length text object is returned. Following is an example: `http.req.cache_control.name(3).contains("some_text")`
`Cache-Control header.IS_INVALID`	Returns a Boolean TRUE if the Cache-Control header is not present in the request or response. Following is an example: `http.req.cache_control.is_invalid`
`Cache-Control header.IS_PRIVATE`	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: `http.req.cache_control.is_private`
`Cache-Control header.IS_PUBLIC`	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: `http.req.cache_control.is_public`
`Cache-Control header.IS_NO_STORE`	Returns a Boolean TRUE if the Cache-Control header has the value No-Store. Following is an example: `http.req.cache_control.is_no_store`

`Cache-Control header.IS_NO_CACHE`	Returns a Boolean TRUE if the Cache-Control header has the value No-Cache. Following is an example: `http.req.cache_control.is_no_cache`
`Cache-Control header.IS_MAX_AGE`	Returns a Boolean TRUE if the Cache-Control header has the value Max-Age. Following is an example: `http.req.cache_control.is_max_age`
`Cache-Control header.IS_MIN_FRESH`	Returns a Boolean TRUE if the Cache-Control header has the value Min-Fresh. Following is an example: `http.req.cache_control.is_min_fresh`
`Cache-Control header.IS_MAX_STALE`	Returns a Boolean TRUE if the Cache-Control header has the value Max-Stale. Following is an example: `http.req.cache_control.is_max_stale`
`Cache-Control header.IS_MUST_REVALIDATE`	Returns a Boolean TRUE if the Cache-Control header has the value Must-Revalidate. Following is an example: `http.req.cache_control.is_must_revalidate`
`Cache-Control header.IS_NO_TRANSFORM`	Returns a Boolean TRUE if the Cache-Control header has the value No-Transform. Following is an example: `http.req.cache_control.is_no_transform`
`Cache-Control header.IS_ONLY_IF_CACHED`	Returns a Boolean TRUE if the Cache-Control header has the value Only-If-Cached. Following is an example: `http.req.cache_control.is_only_if_cached`
`Cache-Control header.IS_PROXY_REVALIDATE`	Returns a Boolean TRUE if the Cache-Control header has the value Proxy-Revalidate. Following is an example: `http.req.cache_control.is_proxy_revalidate`

`Cache-Control header.IS_S_MAXAGE`	Returns a Boolean TRUE if the Cache-Control header has the value S-Maxage. Following is an example: `http.req.cache_control.is_s_maxage`
`Cache-Control header.IS_UNKNOWN`	Returns a Boolean TRUE if the Cache-Control header is of an unknown type. Following is an example: `http.req.cache_control.is_unknown`
`Cache-Control header.MAX_AGE`	Returns the value of the Cache-Control header Max-Age. If this header is absent or invalid, 0 is returned. Following is an example: `http.req.cache_control.max_age.le(3)`
`Cache-Control header.MAX_STALE`	Returns the value of the Cache-Control header Max-Stale. If this header is absent or invalid, 0 is returned. Following is an example: `http.req.cache_control.max_stale.le(3)`
`Cache-Control header.MIN_FRESH`	Returns the value of the Cache-Control header Min-Fresh. If this header is absent or invalid, 0 is returned. Following is an example: `http.req.cache_control.min_fresh.le(3)`
`Cache-Control header.S_MAXAGE`	Returns the value of the Cache-Control header S-Maxage. If this header is absent or invalid, 0 is returned. Following is an example: `http.req.cache_control.s_maxage.eq(2)`

Expressions for Extracting Segments of URLs

You can extract URLs and portions of URLs, such as the host name, or a segment of the URL path. For example, the following expression identifies HTTP requests for image files by extracting image file suffixes from the URL:

```
http.req.url.suffix.eq("jpeg") || http.req.url.suffix.eq("gif")
```

Most expressions for URLs operate on text and are described in Expression Prefixes for Text in HTTP Requests and Responses. This section discusses the GET operation. The GET operation extracts text when used with the following prefixes:

- `HTTP.REQ.URL.PATH`
- `VPN.BASEURL.PATH`
- `VPN.CLIENTLESS_BASEURL.PATH`

The following table describes prefixes for HTTP URLs.

Table 1. Prefixes That Extract URLs

URL Prefix	Description

`HTTP.REQ.URL.PATH.GET(<n>)`	Returns a slash- (“/”) separated list from the URL path. For example, consider the following URL:
	
	`http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1`
	
	The following expression returns dir1 from this URL:
	
	`http.req.url.path.get(1)`
	
	The following expression returns dir2:
	
	`http.req.url.path.get(2)`

HTTP.REQ.URL.PATH.GET_REVERSE(<n>)	Returns a slash- (“/”) separated list from the URL path, starting from the end of the path. For example, consider the following URL: http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1 The following expression returns index.html from this URL: http.req.url.path.get_reverse(0) The following expression returns dir3: http.req.url.path.get_reverse(1)

Expressions for HTTP Status Codes and Numeric HTTP Payload Data Other Than Dates

The following table describes prefixes for numeric values in HTTP data other than dates.

Table 1. Prefixes That Evaluate HTTP Request or Response Length

Prefix	Description
`HTTP.REQ.CONTENT_LENGTH`	Returns the length of an HTTP request as a number. Following is an example: `http.req.content_length < 500`
`HTTP.RES.CONTENT_LENGTH`	Returns the length of the HTTP response as a number. Following is an example: `http.res.content_length <= 1000`
`HTTP.RES.STATUS`	Returns the response status code

HTTP.RES.IS_REDIRECT	Returns a Boolean TRUE if the response code is associated with a redirect. Following are the redirect response codes:
	• 300 (Multiple Choices)
	• 301 (Moved Permanently)
	• 302 (Found)
	• 303 (See Other)
	• 305 (Use Proxy)
	• 307 (Temporary Redirect)
	Note: Status code 304 is not considered a redirect HTTP response status code. Status code 306 is unused.
	In the following example, the rewrite action replaces http in the Location header of an HTTP response with https if the response is associated with an HTTP redirect.
	`add rewrite action redloc replace 'http.res.header("Location").before_regex(re#://#)' '"https"'`
	`add rewrite policy pol1 HTTP.RES.IS_REDIRECT red_location`
	`bind rewrite global pol1 100`

Operations for HTTP, HTML, and XML Encoding and “Safe” Characters

The following operations work with the encoding of HTML data in a request or response and XML data in a POST body.

Table 1. Operations That Evaluate HTML and XML Encoding

peration	Description
_XML_SAFE	Transforms special characters into XML safe format, examples: • A left-pointing angle bracket (<) is converted to • A right-pointing angle bracket (>) is converted to • An ampersand (&) is converted to & This operation safeguards against cross-site scripting of the transformed text is 2048 bytes. This is a read- After applying the transformation, additional operat expression are applied to the selected text. Followin `http.req.url.query.html_xml_safe. conta`
_HEADER_SAFE	Converts all new line ('\n') characters in the input te input to be used safely in HTTP headers. This operation safeguards against response-splitting The maximum length of the transformed text is 2048 operation.

_URL_SAFE	Converts unsafe URL characters to '%xx' values, wher representation of the input character. For example, represented as %26 in URL-safe encoding. The maxin transformed text is 2048 bytes. This is a read-only o Following are URL safe characters. All others are uns • Alpha-numeric characters: a-z, A-Z, 0-9 • Asterix: "*" • Ampersand: "&" • At-sign: "@" • Colon: ":" • Dollar: "$" • Dot: "." • Equals: "=" • Exclamation mark: "!" • Hyphen: "-" • Open and close parentheses: "(", ")" • Plus: "+" • Semicolon: ";" • Single quote: "'" • Slash: "/" • Tilde: "~" • Underscore: "_"
_SAFE	Marks the text as safe without applying any type of e

TEXT_MODE(URLENCODED|NOURLENCODED)	Transforms all %HH encoding in the byte stream. Thi characters (not bytes). By default, a single byte repr ASCII encoding. However, if you specify URLENCODE represent a character.
	
	In the following example, a PREFIX(3) operation sele in a target.
	
	`http.req.url.hostname.prefix(3)`
	
	In the following example, the NetScaler can select u target:
	
	`http.req.url.hostname.set_text_mode(url`

TEXT_MODE(PLUS_AS_SPACE|NO_PLUS_AS_SPACE)	Specifies how to treat the plus character (+). The PL replaces a plus character with white space. For exar “hello+world” becomes “hello world.” The NO_PLUS plus characters as they are.
TEXT_MODE(BACKSLASH_ENCODED|NO_BACKSLASH_ENCODED)	Specifies whether or not backslash decoding is perfo represented by <text>. If BACKSLASH_ENCODED is specified, the SET_TEX performs the following operations on the text object • All occurrences of “\XXX” will be replaced with t XXX represents a number in the octal system and equivalent of XXX). The valid range of octal valu encoding is 0 to 377. For example, the encoded "http\072//" will both be decoded to "http://", v ASCII equivalent of the octal value “72”. • All occurrences of “\xHH” will be replaced with represents a number in the hexadecimal system equivalent of HH. For example, the encoded tex decoded to "http://", where the colon (:) is the hexadecimal value “3a“. • All occurrences of “\\uWWXX” will be replaced v sequence “YZ” (Where WW and XX represent tw values and Y and Z represent their ASCII equivale respectively. For example, the encoded text "ht "http%u003a//" will both be decoded to "http:// are two hexadecimal values and the colon (:) an represent their ASCII equivalents respectively. • All occurrences of "\b", "\n", "\t", "\f", and "\r" are corresponding ASCII characters. If NO_BACKSLASH_ENCODED is specified, backslash performed on the text object.

TEXT_MODE(BAD_ENCODE_RAISE_UNDEF|NO_BAD_ENCODE_RAISE_UNDEF)	Performs the associated undefined action if either th BACKSLASH_ENCODED mode is set and bad encodin specified encoding mode is encountered in the text <text>. If NO_BAD_ENCODE_RAISE_UNDEF is specified, the action will not be performed when bad encoding is e object represented by<text>.

Expressions for TCP, UDP, and VLAN Data

TCP and UDP data take the form of a string or a number. For expression prefixes that return string values for TCP and UDP data, you can apply any text-based operations. For more information, see Default Syntax Expressions: Evaluating Text.

For expression prefixes that return numeric value, such as a source port, you can apply an arithmetic operation. For more information, see Basic Operations on Expression Prefixes and Compound Operations for Numbers.

The following table describes prefixes that extract TCP and UDP data.

Table 1. Prefixes That Extract TCP and UDP Data

GET Operation	Description
`CLIENT.TCP.PAYLOAD(<integer>)`	Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the <integer> argument. You can apply any text-based operation to this prefix.
`CLIENT.TCP.SRCPORT`	Returns the ID of the current packet's source port as a number.
`CLIENT.TCP.DSTPORT`	Returns the ID of the current packet's destination port as a number.
`CLIENT.TCP.OPTIONS`	Returns the TCP options set by the client. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The `COUNT`, `TYPE(<type>)`, and `TYPE_NAME(<m>)` operators can be used with this prefix. For the TCP options set by the server, see the `SERVER.TCP.OPTIONS` prefix.
`CLIENT.TCP.OPTIONS.COUNT`	Returns the number of TCP options that the client has set.

`CLIENT.TCP.OPTIONS.TYPE(<type>)`	Returns the value of the TCP option whose type (or *option kind*) is specified as the argument. The value is returned as a string of bytes in big endian format (or *network byte order*). **Parameters:** `type` - Type value
`CLIENT.TCP.OPTIONS.TYPE_NAME(<m>)`	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use `CLIENT.TCP.OPTIONS.TYPE(<type>)`. For other TCP options, you must use `CLIENT.TCP.OPTIONS.TYPE(<type>)`. **Parameters:** `m` - TCP option enumeration constant
`CLIENT.TCP.REPEATER_OPTION.EXISTS`	Returns a Boolean TRUE if Repeater TCP options exist.
`CLIENT.TCP.REPEATER_OPTION.IP`	Returns the branch repeater's IPv4 address from the Repeater TCP options.
`CLIENT.TCP.REPEATER_OPTION.MAC`	Returns the branch repeater's MAC address from the Repeater TCP options.
`CLIENT.UDP.DNS.DOMAIN`	Returns the DNS domain name.
`CLIENT.UDP.DNS.DOMAIN.EQ("<hostname>")`	Returns a Boolean TRUE if the domain name matches the <hostname> argument. The comparison is case insensitive. Following is an example: `client.udp.dns.domain.eq("www.mycompany.com")`
`CLIENT.UDP.DNS.IS_AAAAREC`	Returns a Boolean TRUE if the record type is AAAA. These types of records indicate an IPv6 address in forward lookups.
`CLIENT.UDP.DNS.IS_ANYREC`	Returns a Boolean TRUE if it is of any record type.
`CLIENT.UDP.DNS.IS_AREC`	Returns a Boolean TRUE if the record is type A. Type A records provide the host address.
`CLIENT.UDP.DNS.IS_CNAMEREC`	Returns a Boolean TRUE if the record is of type CNAME. In systems that use multiple names to identify a resource, there is one canonical name and a number of aliases. The CNAME provides the canonical name.

CLIENT.UDP.DNS.IS_MXREC	Returns a Boolean TRUE if the record is of type MX (mail exchanger). This DNS record describes a priority and a host name. The MX records for the same domain name specify the email servers in the domain and the priority for each server.
CLIENT.UDP.DNS.IS_NSREC	Returns a Boolean TRUE if the record is of type NS. This is a name server record that includes a host name with an associated A record. This enables locating the domain name that is associated with the NS record.
CLIENT.UDP.DNS.IS_PTRREC	Returns a Boolean TRUE if the record is of type PTR. This is a domain name pointer and is often used to associate a domain name with an IPv4 address.
CLIENT.UDP.DNS.IS_SOAREC	Returns a Boolean TRUE if the record is of type SOA. This is a start of authority record.
CLIENT.UDP.DNS.IS_SRVREC	Returns a Boolean TRUE if the record is of type SRV. This is a more general version of the MX record.
CLIENT.UDP.DSTPORT	Returns the numeric ID of the current packet's UDP destination port.
CLIENT.UDP.SRCPORT	Returns the numeric ID of the current packet's UDP source port.
CLIENT.UDP.RADIUS	Returns RADIUS data for the current packet.
CLIENT.UDP.RADIUS.ATTR_TYPE(<type>)	Returns the value for the attribute type specified as the argument.
CLIENT.UDP.RADIUS.USERNAME	Returns the RADIUS user name.
CLIENT.TCP.MSS	Returns the maximum segment size (MSS) for the current connection as a number.
CLIENT.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the NetScaler.
SERVER.TCP.DSTPORT	Returns the numeric ID of the current packet's destination port.
SERVER.TCP.SRCPORT	Returns the numeric ID of the current packet's source port.
SERVER.TCP.OPTIONS	Returns the TCP options set by the server. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this prefix. For the TCP options set by the client, see the CLIENT.TCP.OPTIONS prefix.
SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options that the server has set.

SERVER.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or *option kind*) is specified as the argument. The value is returned as a string of bytes in big endian format (or *network byte order*). **Parameters:** type - Type value
SERVER.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). **Parameters:** m - TCP option enumeration constant
SERVER.VLAN	Operates on the VLAN through which the current packet entered the NetScaler.
SERVER.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the NetScaler.

XPath and JSON Expressions

The default syntax expression engine supports expressions for evaluating and retrieving data from XML and JavaScript Object Notation (JSON) files. This enables you to find specific nodes in an XML or JSON document, determine if a node exists in the file, locate nodes in XML contexts (for example, nodes that have specific parents or a specific attribute with a given value), and return the contents of such nodes. Additionally, you can use XPath expressions in rewrite expressions.

The default syntax expression implementation for XPath comprises a default syntax expression prefix (such as "`HTTP.REQ.BODY`") that designates XML text and the XPATH operator that takes the XPath expression as its argument.

JSON files are either a collection of name/value pairs or an ordered list of values. You can use the XPATH_JSON operator, which takes an XPath expression as its argument, to process JSON files.

Table 1. XPath and JSON Expression Prefixes That Return Text

XPath Prefix	Description
`<text>.XPATH(xpathex)`	Operate on an XML file and return a Boolean value. For example, the following expression returns a Boolean TRUE if a node calle "creator" exists under the node "Book" within the first 1000 bytes of the XM `HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%)` Parameters: xpathex - XPath Boolean expression
`<text>.XPATH(xpathex)`	Operate on an XML file and return a value of data type "double." For example, the following expression converts the string "36" (a price valu value of data type "double" if the string is in the first 1000 bytes of the XML `HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%)` Parameters: xpathex - XPath numeric expression

`<text>.XPATH(xpathex)`	Operate on an XML file and return a node-set or a string. Node-sets are conv to corresponding strings by using the standard XPath string conversion routin For example, the following expression selects all the nodes that are enclose “/Book/creator” (a node-set) in the first 1000 bytes of the body: `HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%)` Parameters: xpathex - XPath expression
`<text>.XPATH_JSON(xpathex)`	Operate on a JSON file and return a Boolean value. For example, {consider the following JSON file: { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } } The following expression operates on the JSON file and returns a Boolean TR the JSON file contains a node named “creator,” whose parent node is “Book the first 1000 bytes: `HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%` Parameters: xpathex - XPath Boolean expression
`<text>.XPATH_JSON(xpathex)`	Operate on a JSON file and return a value of data type “double.” For example, consider the following JSON file: { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>', "price":"3 The following expression operates on the JSON file and converts the string “ a value of data type “double” if the string is present in the first 1000 bytes o JSON file. `HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%)` Parameters: xpathex - XPath numeric expression

<text>.XPATH_JSON(xpathex)	Operate on a JSON file and return a node-set or a string. Node-sets are conv to corresponding strings by using the standard XPath string conversion routin For example, consider the following JSON file: { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } } The following expression selects all the nodes that are enclosed by “/Book" (node-set) in the first 1000 bytes of the body of the JSON file and returns the corresponding string value, which is "<name><title>": HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%) Parameters: xpathex - XPath expression
<text>.XPATH_JSON_WITH_MARKUP(xpathex)	Operate on an XML file and return a string that contains the entire portion o document for the result node, including markup such as including the enclos element tags. For example, consider the following JSON file: {"Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } } The following expression operates on the JSON file and selects all the nodes are enclosed by “/Book/creator" in the first 1000 bytes of the body, which is “creator:{ person:{ name:'<name>' } }.” HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creat The portion of the JSON body that is selected by the expression is marked fo further processing. Parameters: xpathex - XPath expression
<text>.XPATH_WITH_MARKUP(xpathex)	Operate on an XML file and return a string that contains the entire portion o document for the result node, including markup such as including the enclos element tags. For example, the following expression operates on an XML file and selects al nodes enclosed by “/Book/creator" in the first 1000 bytes of the body. HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%) The portion of the JSON body that is selected by the expression is marked fo further processing. Parameters: xpathex - XPath expression

Encrypting and Decrypting XML Payloads

You can use the XML_ENCRYPT() and XML_DECRYPT() functions in default syntax expressions to encrypt and decrypt, respectively, XML data. These functions conform to the W3C XML Encryption standard defined at http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/. XML_ENCRYPT() and XML_DECRYPT() support a subset of the XML Encryption specification. In the subset, data encryption uses a bulk cipher method (RC4, DES3, AES128, AES192, or AES256), and an RSA public key is used to encrypt the bulk cipher key.

Note: If you want to encrypt and decrypt text in a payload, you must use the ENCRYPT and DECRYPT functions. For more information about these functions, see Encrypting and Decrypting Text.

The XML_ENCRYPT() and XML_DECRYPT() functions are not dependent on the encryption/decryption service that is used by the ENCRYPT and DECRYPT commands for text. The cipher method is specified explicitly as an argument to the XML_ENCRYPT() function. The XML_DECRYPT() function obtains the information about the specified cipher method from the <xenc:EncryptedData> element. Following are synopses of the XML encryption and decryption functions:

- **XML_ENCRYPT(<certKeyName>, <method> [, <flags>])**. Returns an <xenc:EncryptedData> element that contains the encrypted input text and the encryption key, which is itself encrypted by using RSA.
- **XML_DECRYPT(<certKeyName>)**. Returns the decrypted text from the input <xenc:EncryptedData> element, which includes the cipher method and the RSA-encrypted key.

Note: The <xenc:EncryptedData> element is defined in the W3C XML Encryption specification.

Following are descriptions of the arguments:

certKeyName

Selects an X.509 certificate with an RSA public key for XML_ENCRYPT() or an RSA private key for XML_DECRYPT(). The certificate key must have been previously created by an add ssl certKey command.

method

Specifies which cipher method to use for encrypting the XML data. Possible values: RC4, DES3, AES128, AES192, AES256.

flags

A bitmask specifying the following optional key information (<ds:KeyInfo>) to be included in the <xenc:EncryptedData> element that is generated by XML_ENCRYPT():

-

1 - Include a `KeyName` element with the `certKeyName`. The element is `<ds:KeyName>`.

- **2** - Include a `KeyValue` element with the RSA public key from the certificate. The element is `<ds:KeyValue>`.
- **4** - Include an `X509IssuerSerial` element with the certificate serial number and issuer DN. The element is `<ds:X509IssuserSerial>`.
- **8** - Include an `X509SubjectName` element with the certificate subject DN. The element is `<ds:X509SubjectName>`.
- **16** - Include an `X509Certificate` element with the entire certificate. The element is `<ds:X509Certificate>`.

Using the *XML_ENCRYPT( )* and *XML_DECRYPT( )* Functions in Expressions

The XML encryption feature uses SSL certificate-key pairs to provide X.509 certificates (with RSA public keys) for key encryption and RSA private keys for key decryption. Therefore, before you use the `XML_ENCRYPT( )` function in an expression, you must create an SSL certificate-key pair. The following command creates an SSL certificate-key pair, `my-certkey`, with the X.509 certificate, `my-cert.pem`, and the private key file, `my-key.pem`.

```
add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem
-passcrypt kxPeMRYnitY=
```

The following CLI commands create rewrite actions and policies for encrypting and decrypting XML content.

```
add rewrite action my-xml-encrypt-action replace
"HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp%/%)"
"HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp%/%).XML_ENCRYPT(\"my-certkey\",
AES256, 31)" -bypassSafetyCheck YES

add rewrite action my-xml-decrypt-action replace
"HTTP.REQ.BODY(10000).XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)"
"HTTP.REQ.BODY(10000).XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT(\"my
-bypassSafetyCheck YES

add rewrite policy my-xml-encrypt-policy
"HTTP.REQ.URL.CONTAINS(\"xml-encrypt\")" my-xml-encrypt-action

add rewrite policy my-xml-decrypt-policy
"HTTP.REQ.BODY(10000).XPATH(xp%boolean(//xenc:EncryptedData)%)"
my-xml-decrypt-action
```

```
bind rewrite global my-xml-encrypt-policy 30

bind rewrite global my-xml-decrypt-policy 30
```

In the above example, the rewrite action `my-xml-encrypt-action` encrypts the entire XML document (`XPATH_WITH_MARKUP(xp%/%)`) in the request by using the AES-256 bulk encryption method and the RSA public key from `my-certkey` to encrypt the bulk encryption key. The action replaces the document with an `<xenc:EncryptedData>` element containing the encrypted data and an encrypted key. The flags represented by 31 include all of the optional `<ds:KeyInfo>` elements.

The action `my-xml-decrypt-action` decrypts the first `<xenc:EncryptedData>` element in the response (`XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)`). This requires the prior addition of the `xenc` XML namespace by use of the following CLI command:

```
add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc#
```

The `my-xml-decrypt-action` action uses the RSA private key in `my-certkey` to decrypt the encrypted key and then uses the bulk encryption method specified in the element to decrypt the encrypted contents. Finally, the action replaces the encrypted data element with the decrypted content.

The rewrite policy `my-xml-encrypt-policy` applies `my-xml-encrypt-action` to requests for URLs containing `xml-encrypt`. The action encrypts the entire response from a service configured on the NetScaler appliance.

The rewrite policy `my-xml-decrypt-policy` applies `my-xml-decrypt-action` to requests that contain an `<xenc:EncryptedData>` element (`(XPATH(xp%//xenc:EncryptedData%)` returns a non-empty string). The action decrypts the encrypted data in requests that are bound for a service configured on the NetScaler appliance.

Default Syntax Expressions: Parsing SSL Certificates

You can use default syntax expressions to evaluate X.509 Secure Sockets Layer (SSL) client certificates. A client certificate is an electronic document that can be used to authenticate a user's identity. A client certificate contains (at a minimum) version information, a serial number, a signature algorithm ID, an issuer name, a validity period, a subject (user) name, a public key, and signatures.

You can examine both SSL connections and data in client certificates. For example, you may want to send SSL requests that use low-strength ciphers to a particular load balancing virtual server farm. The following command is an example of a Content Switching policy that parses the cipher strength in a request and matches cipher strengths that are less than or equal to 40:

```
add cs policy p1 -rule "client.ssl.cipher_bits.le(40)"
```

As another example, you can configure a policy that determines whether a request contains a client certificate:

```
add cs policy p2 -rule "client.ssl.client_cert EXISTS"
```

Or, you might want to configure a policy that examines particular information in a client certificate. For example, the following policy verifies that the certificate has one or more days before expiration:

```
add cs policy p2 -rule "client.ssl.client_cert exists &&
client.ssl.client_cert.days_to_expire.ge(1)"
```

Note: For information on parsing dates and times in a certificate, see Format of Dates and Times in an Expression and Expressions for SSL Certificate Dates.

Prefixes for Text-Based SSL and Certificate Data

The following table describes expression prefixes that identify text-based items in SSL transactions and client certificates.

Table 1. Prefixes That Return Text or Boolean Values for SSL and Client Certificate Data

Prefix	Description
`CLIENT.SSL.CLIENT_CERT`	Returns the SSL client certificate in the current SSL transaction.
`CLIENT.SSL.CLIENT_CERT.TO_PEM`	Returns the SSL client certificate in binary format.
`CLIENT.SSL.CIPHER_EXPORTABLE`	Returns a Boolean TRUE if the SSL cryptographic SSL cryptographic cipher is exportable.
`CLIENT.SSL.CIPHER_NAME`	Returns the name of the SSL Cipher if invoked from an SSL connection, and a NULL string if invoked from a non-SSL connection.
`CLIENT.SSL.IS_SSL`	Returns a Boolean TRUE if the current connection is SSL-based.

Prefixes for Numeric Data in SSL Certificates

The following table describes prefixes that evaluate numeric data other than dates in SSL certificates. These prefixes can be used with the operations that are described in Basic Operations on Expression Prefixes and Compound Operations for Numbers.

Table 1. Prefixes That Evaluate Numeric Data Other Than Dates in SSL Certificates

Prefix	Description
`CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE`	Returns the number of days that the certificate is valid, or returns -1 for expired certificates.
`CLIENT.SSL.CLIENT_CERT.PK_SIZE`	Returns the size of the public key used in the certificate.
`CLIENT.SSL.CLIENT_CERT.VERSION`	Returns the version number of the certificate. If the connection is not SSL-based, returns zero (0).
`CLIENT.SSL.CIPHER_BITS`	Returns the number of bits in the cryptograhic key. Returns 0 if the connection is not SSL-based.
`CLIENT.SSL.VERSION`	Returns a number that represents the SSL protocol version, as follows: • **0.** The transaction is not SSL-based. • **0x002.** The transaction is SSLv2. • **0x300.** The transaction is SSLv3. • **0x301.** The transaction is TLSv1.

Note: For expressions related to expiration dates in a certificate, see Expressions for SSL Certificate Dates.

Expressions for SSL Certificates

You can parse SSL certificates by configuring expressions that use the following prefix:

```
CLIENT.SSL.CLIENT_CERT
```

This section discusses the expressions that you can configure for certificates, with the exception of expressions that examine certificate expiration. Time-based operations are described in Default Syntax Expressions: Working with Dates, Times, and Numbers.

The following table describes operations that you can specify for the `CLIENT.SSL.CLIENT_CERT prefix`.

Table 1. Operations That Can Be Specified with the CLIENT.SSL.CLIENT_CERT Prefix

SSL Certificate Operation	Description
`<certificate>.EXISTS`	Returns a Boolean TRUE if the client has an SSL certificate.
`<certificate>.ISSUER`	Returns the Distinguished Name (DN) of the Issuer in the certificate as a name-value list. An equals sign (“=”) is the delimiter for the name and the value, and the slash (“/”) is the delimiter that separates the name-value pairs. Following is an example of the returned DN: `/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/email`

`<certificate>.ISSUER.IGNORE_EMPTY_ELEMENTS`	Returns the Issuer and ignores the empty elements in a name-value list. For example, consider the following:
	
	`Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com`
	
	The following Rewrite action returns a count of 6 based on the preceding Issuer definition:
	
	`sh rewrite action insert_ssl_header`
	
	`Name: insert_ssl`
	
	`Operation: insert_http_header Target:Cert-Issuer`
	
	`Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT`
	
	However, if you change the value to the following, the returned count is 9:
	
	`CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT`

<certificate>.AUTH_KEYID	Returns a string that contains the Authority Key Identifier extension of the X.509 V3 certificate.
<certificate>.AUTH_KEYID.SERIAL_NUMBER	Returns the serial number in the Authority Key Identifier as a blob.
<certificate>.AUTH_KEYID.EXIST	Returns TRUE if the certificate contains an Authority Key Identifier extension.
<certificate>.AUTH_KEYID.ISSUER_NAME	Returns the Issuer Distinguished Name in the certificate as a name-value list. An equals sign (“=”) is the delimiter for the name and the value, and the slash (“/”) is the delimiter that separates the name-value pairs. Following is an example: /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/email
<certificate>.AUTH_KEYID.ISSUER_NAME.IGNORE_EMPTY_ELEMENTS	Returns the Issuer Distinguished Name in the certificate as a name-value list and ignores the empty elements in the list. For example, the following name-value list has an empty element following “a=10”: a=10;;b=11; ;c=89 The element following b=11 is not considered an empty element.
<certificate>.AUTH_KEYID.KEYID	Returns the keyidentifier field of the Authority Key Identifier as a blob.
<certificate>.CERT_POLICY	Returns a string that contains the client certificate policy. Note that this represents a sequence of certificate policies.

<certificate>.KEY_USAGE(string)	Returns a boolean value to indicate whether the specified key usage extension bit value in the X.509 certificate is set. The string argument specifies which bit is checked. Following are valid arguments: • **DIGITAL_SIGNATURE.** Returns TRUE if the digital signature bit is set; otherwise, it returns FALSE. • **NONREPUDIATION.** Returns TRUE if the nonrepudiation bit is set; otherwise, it returns FALSE. • **KEYENCIPHERMENT.** Returns TRUE if the key encipherment bit is set; otherwise, it returns FALSE. • **DATAENCIPHERMENT.** Returns TRUE if the data encipherment bit is set; otherwise, it returns FALSE. • **KEYAGREEMENT.** Returns TRUE if the key agreement bit is set; otherwise, it returns FALSE. • **KEYCERTSIGN.** Returns TRUE if the key cert sign bit is set; otherwise, it returns FALSE. • **CRLSIGN.** Returns TRUE if the CRL bit is set; otherwise, it returns FALSE. • **ENCIPHERONLY.** Returns TRUE if the encipher only bit is set; otherwise, it returns FALSE. • **DECIPHERONLY.** Returns TRUE if the decipher only bit is set; otherwise, it returns FALSE.
<certificate>.PK_ALGORITHM	Returns the name of the public key algorithm used by the certificate.
<certificate>.PK_SIZE	Returns the size of the public key used in the certificate.
<certificate>.SERIALNUMBER	Returns the serial number of the client certificate. If this is a non-SSL transaction or there is an error in the certificate, this operation returns an empty string.
<certificate>.SIGNATURE_ALGORITHM	Returns the cryptographic algorithm used by the CA to sign this certificate.
<certificate>.SUBJECT	Returns the Distinguished Name of the Subject as a name-value. An equals sign (“=”) separates names and values and a slash (“/”) delimits name-value pairs. Following is an example: /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/email

<certificate>.SUBJECT.IGNORE_EMPTY_ELEMENTS	Returns the Subject as a name-value list, but ignores the empty elements in the list. For example, consider the following:
	
	`Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com`
	
	The following Rewrite action returns a count of 6 based on the preceding Issuer definition:
	
	`sh rewrite action insert_ssl_header`
	
	`Name: insert_ssl`
	
	`Operation: insert_http_header Target:Cert-Issuer`
	
	`Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT`
	
	However, if you change the value to the following, the returned count is 9:
	
	`CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT`

<certificate>.SUBJECT.KEYID	Returns the Subject KeyID of the client certificate. If there is no Subject KeyID, this operation returns a zero-length text object.

Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs

You can use default syntax expression prefixes that return IPv4 and IPv6 addresses, MAC addresses, IP subnets, useful client and server data such as the throughput rates at the interface ports (Rx, Tx, and RxTx), and the IDs of the VLANs through which packets are received. You can then use various operators to evaluate the data that is returned by these expression prefixes.

Expressions for IP Addresses and IP Subnets

You can use default syntax expressions to evaluate addresses and subnets that are in Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) format. Expression prefixes for IPv6 addresses and subnets include IPv6 in the prefix. Expression prefixes for IPv4 addresses and subnets include IP in the prefix. Following is an example of an expression that identifies whether a request has originated from a particular IPv4 subnet.

```
client.ip.src.in_subnet(147.1.0.0/16)
```

Following are two examples of Rewrite policies that examine the subnet from which the packet is received and perform a rewrite action on the Host header. With these two policies configured, the rewrite action that is performed depends on the subnet in the request. These two policies evaluate IP addresses that are in the IPv4 address format.

```
add rewrite action URL1-rewrite-action replace "http.req.header(\"Host\")" "\"www.mycompany1.com\""
add rewrite policy URL1-rewrite-policy "http.req.header(\"Host\").contains(\"www.test1.com\") && client.ip.
add rewrite action URL2-rewrite-action replace "http.req.header(\"Host\")" "\"www.mycompany2.com\""
add rewrite policy URL2-rewrite-policy "http.req.header(\"Host\").contains(\"www.test2.com\") && client.ip.
```

Note: The preceding examples are commands that you type at the NetScaler command-line interface (CLI) and, therefore, each quotation mark must be preceded by a backslash (\). For more information, see Configuring Default Syntax Expressions in a Policy.

Prefixes for IPV4 Addresses and IP Subnets

The following table describes prefixes that return IPv4 addresses and subnets, and segments of IPv4 addresses. You can use numeric operators and operators that are specific to IPv4 addresses with these prefixes. For more information about numeric operations, see Basic Operations on Expression Prefixes and Compound Operations for Numbers.

Table 1. Prefixes That Evaluate IP and MAC Addresses

Prefix	Description
`CLIENT.IP.SRC`	Returns the source IP of the current packet as an IP address or as a number.
`CLIENT.IP.DST`	Returns the destination IP of the current packet as an IP address or as a number.
`SERVER.IP.SRC`	Returns the source IP of the current packet as an IP address or as a number.
`SERVER.IP.DST`	Returns the destination IP of the current packet as an IP address or as a number.

Operations for IPV4 Addresses

The following table describes the operators that can be used with prefixes that return an IPv4 address.

Table 1. Operations on IPV4 Addresses

Prefix	Description
`<ip address>.EQ(<address>)`	Returns a Boolean TRUE if the IP address value is same as the <address> argument. The following example checks whether the client's destination IP address is equal to 10.100.10.100: `client.ip.dst.eq(10.100.10.100)`

`<ip address>.GET1. . .GET4`	Returns a portion of an IP address as a numeric value. For example, if the IP address value is 10.100.200.1, the following is returned:
	
	`client.ip.src.get1 Returns 10`
	
	`client.ip.src.get2 returns 100`
	
	`client.ip.src.get3 returns 200`

`<ip address>.IN_SUBNET(<subnet>)`	Returns a Boolean TRUE if the <subnet> argument matches the subnet of th address value. For example, the following determines whether the client's destination IP address subnet is 10.100.10.100/18: `client.ip.dst.eq(10.100.10.100/18)`
`<ip address>.SUBNET(<n>)`	Returns the IP address after applying the subnet mask specified as the argu The subnet mask can take values between 0 and 32. For example: `CLIENT.IP.SRC.SUBNET(24) returns 192.168.1.0 if the IP address represented by the prefix is 192.168.1.[0-255].`
`<ip address>.IS_IPV6`	Returns a Boolean TRUE if this is an Internet Protocol version 6 (IPv6) host f the client or server. Following is an example: `client.ip.src.is_ipv6`
`<ip address>.MATCHES(<hostname>)`	Returns a Boolean TRUE if the IP address for the host specified in <hostnam matches the current IP address. The <hostname> cannot exceed 255 charac

`<ip address>.MATCHES_LOCATION(<location>)`	Returns a Boolean TRUE if the location of the IP address matches the <locat argument. The Location string can take the following form: *qual1.qual2.qual3.qual4.qual5.qual6*, for example: NorthAmeria.CA.* Following is an example: `client.ip.src.matches_location(\"Europe.GB.17.London.*.*\`

About IPv6 Expressions

The IPv6 address format allows more flexibility than the older IPv4 format. IPv6 addresses are in the hexadecimal format (RFC 2373). In the following examples, Example 1 is an IPv6 address, Example 2 is a URL that includes the IPv6 address, and Example 3 includes the IPv6 address and a port number.

Example 1:

9901:0ab1:22a2:88a3:3333:4a4b:5555:6666

Example 2:

http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/

Example 3:

https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/

In Example 3, the brackets separate the IP address from the port number (8080).

Note that you can only use the '+' operator to combine IPv6 expressions with other expressions. The output is a concatenation of the string values that are returned from the individual expressions. You cannot use any other arithmetic operator with an IPv6 expression. The following syntax is an example:

client.ipv6.src + server.ip.dst

For example, if the client source IPv6 address is ABCD:1234::ABCD, and the server destination IPv4 address is 10.100.10.100, the preceding expression returns "ABCD:1234::ABCD10.100.10.100".

Note that when the NetScaler appliance receives an IPv6 packet, it assigns a temporary IPv4 address from an unused IPv4 address range and changes the source address of the packet to this temporary address. At response time, the outgoing packet's source address is replaced with the original IPv6 address.

Note: You can combine an IPv6 expression with any other expression except an expression that produces a Boolean result.

Expression Prefixes for IPv6 Addresses

The IPv6 addresses that are returned by the expression prefixes in the following table can be treated as text data. For example, the prefix client.ipv6.dst returns the destination IPv6 address as a string that can be evaluated as text.

The following table describes expression prefixes that return an IPv6 address.

Table 1. IPv6 Expression Prefixes That Return Text

Prefix	Description
`CLIENT.IPV6`	Operates on the IPv6 address in with the current packet.
`CLIENT.IPV6.DST`	Returns the IPv6 address in the destination field of the IP header.

`CLIENT.IPV6.SRC`	Returns the IPv6 address in the source field of the IP header. Following are examples:
	
	`client.ipv6.src.in_subnet(2007::2008/64)`
	
	`client.ipv6.src.get1.le(2008)`

`SERVER.IPV6`	Operates on the IPv6 address in with the current packet.
`SERVER.IPV6.DST`	Returns the IPv6 address in the destination field of the IP header.

`SERVER.IPV6.SRC`	Returns the IPv6 address in the source field of the IP header. Following are examples:
	
	`server.ipv6.src.in_subnet(2007::2008/64)`
	
	`server.ipv6.src.get1.le(2008)`

Operations for IPV6 Prefixes

The following table describes the operators that can be used with prefixes that return an IPv6 address:

Table 1. Operations That Evaluate IPv6 Addresses

IPv6 Operation	Description
`<ipv6>.EQ(<IPv6_address> )`	Returns a Boolean TRUE if the IP address value is same as the <IPv6_address> argument. Following is an example: `client.ipv6.dst.eq(ABCD:1234::ABCD)`
`<ipv6>.GET1. . .GET8`	Returns a segment of an IPv6 address as a number. The following example expressions retrieve segments from the ipv6 address 1000:1001:CD10:0000:0000:89AB:4567:CDEF: • `client.ipv6.dst.get5` extracts 0000, which is the fifth set of bits in the address. • `client.ipv6.dst.get6 extracts 89AB.` • `client.ipv6.dst.get7` extracts 4567. You can perform numeric operations on these segments. Note that you cannot perform numeric operations when you retrieve an entire IPv6 address. This is because expressions that return an entire IPv6 address, such as `CLIENT.IPV6.SRC`, return the address in text format.
`<ipv6>.IN_SUBNET(<subnet>)`	Returns a Boolean TRUE if the IPv6 address value is in the subnet specified by the <subnet> argument. Following is an example: `client.ipv6.dst.eq(1000:1001:CD10:0000:0000:89AB:4567:CDEF/60)`
`<ipv6>.IS_IPV4`	Returns a Boolean TRUE if this is an IPv4 client, and returns a Boolean FALSE if it is not.
`<ipv6>.SUBNET(<n>)`	Returns the IPv6 address after applying the subnet mask specified as the argument. The subnet mask can take values between 0 and 128. For example: `CLIENT.IPV6.SRC.SUBNET(24)`

Expressions for MAC Addresses

A MAC address consists of colon-delimited hexadecimal values in the format ##:##:##:##:##:##, where each “#” represents either a number from 0 through 9 or a letter from A through F. Default syntax expression prefixes and operators are available for evaluating source and destination MAC addresses.

Prefixes for MAC Addresses

The following table describes prefixes that return MAC addresses.

Table 1. Prefixes That Evaluate MAC Addresses

Prefix	Description
`client.ether.dstmac`	Returns the MAC address in the destination field of the Ethernet header.
`client.ether.srcmac`	Returns the MAC address in the source field of the Ethernet header.

Operations for MAC Addresses

The following table describes the operators that can be used with prefixes that return a MAC address.

Table 1. Operations on MAC Addresses

Prefix	Description
`<mac address>.EQ(<address>)`	Returns a Boolean TRUE if the MAC address value is same as the <address> argument.
`<mac address>.GET1. . .GET4`	Returns a numeric value extracted from the segment of the MAC address that is specified in the GET operation. For example, if the MAC address is 12:34:56:78:9a:bc, the following returns 34: `client.ether.dstmac.get2`

Expressions for Numeric Client and Server Data

The following table describes prefixes for working with numeric client and server data, including throughput, port numbers, and VLAN IDs.

Table 1. Prefixes That Evaluate Numeric Client and Server Data

Prefix	Description
`client.interface.rxthroughput`	Returns an integer representing the raw received traffic throughput in kilobytes per second (KBps) for the previous seven seconds.
`client.interface.txthroughput`	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
`client.interface.rxtxthroughput`	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
`server.interface.rxthroughput`	Returns an integer representing the raw received traffic throughput in KBps for the previous seven seconds.
`server.interface.txthroughput`	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
`server.interface.rxtxthroughput`	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
`server.vlan.id`	Returns a numeric ID of the VLAN through which the current packet entered the NetScaler.
`client.vlan.id`	Returns a numeric ID for the VLAN through which the current packet entered the NetScaler.

Default Syntax Expressions: DataStream

The policy infrastructure on the Citrix® NetScaler® appliance includes expressions that you can use to evaluate and process database server traffic when the appliance is deployed between a farm of application servers and their associated database servers.

Expressions for the MySQL Protocol

The following expressions evaluate traffic associated with MySQL database servers. You can use the request-based expressions (expressions that begin with `MYSQL.CLIENT` and `MYSQL.REQ`) in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with `MYSQL.RES`) to evaluate server responses to user-configured health monitors.

- **`MYSQL.CLIENT`.** Operates on the client properties of a MySQL connection.
- **`MYSQL.CLIENT.CAPABILITIES`.** Returns the set of flags that the client has set in the capabilities field of the handshake initialization packet during authentication. Examples of the flags that are set are `CLIENT_FOUND_ROWS`, `CLIENT_COMPRESS`, and `CLIENT_SSL`.
- **`MYSQL.CLIENT.CHAR_SET`.** Returns the enumeration constant assigned to the character set that the client uses. The `EQ(<m>)` and `NE(<m>)` operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the character set enumeration constants:
  - `LATIN2_CZECH_CS`
  - `DEC8_SWEDISH_CI`
  - `CP850_GENERAL_CI`
  - `GREEK_GENERAL_CI`
  - `LATIN1_GERMAN1_CI`
  - `HP8_ENGLISH_CI`
  - `KOI8R_GENERAL_CI`
  - `LATIN1_SWEDISH_CI`
  - `LATIN2_GENERAL_CI`
  - `SWE7_SWEDISH_CI`
  - `ASCII_GENERAL_CI`
  - `CP1251_BULGARIAN_CI`
  - `LATIN1_DANISH_CI`
  - `HEBREW_GENERAL_CI`
  - `LATIN7_ESTONIAN_CS`

- LATIN2_HUNGARIAN_CI
- KOI8U_GENERAL_CI
- CP1251_UKRAINIAN_CI
- CP1250_GENERAL_CI
- LATIN2_CROATIAN_CI
- CP1257_LITHUANIAN_CI
- LATIN5_TURKISH_CI
- LATIN1_GERMAN2_CI
- ARMSCII8_GENERAL_CI
- UTF8_GENERAL_CI
- CP1250_CZECH_CS
- CP866_GENERAL_CI
- KEYBCS2_GENERAL_CI
- MACCE_GENERAL_CI
- MACROMAN_GENERAL_CI
- CP852_GENERAL_CI
- LATIN7_GENERAL_CI
- LATIN7_GENERAL_CS
- MACCE_BIN
- CP1250_CROATIAN_CI
- LATIN1_BIN
- LATIN1_GENERAL_CI
- LATIN1_GENERAL_CS
- CP1251_BIN
- CP1251_GENERAL_CI
- CP1251_GENERAL_CS
- MACROMAN_BIN
- CP1256_GENERAL_CI

- CP1257_BIN
- CP1257_GENERAL_CI
- ARMSCII8_BIN
- ASCII_BIN
- CP1250_BIN
- CP1256_BIN
- CP866_BIN
- DEC8_BIN
- GREEK_BIN
- HEBREW_BIN
- HP8_BIN
- KEYBCS2_BIN
- KOI8R_BIN
- KOI8U_BIN
- LATIN2_BIN
- LATIN5_BIN
- LATIN7_BIN
- CP850_BIN
- CP852_BIN
- SWE7_BIN
- UTF8_BIN
- GEOSTD8_GENERAL_CI
- GEOSTD8_BIN
- LATIN1_SPANISH_CI
- UTF8_UNICODE_CI
- UTF8_ICELANDIC_CI
- UTF8_LATVIAN_CI
- UTF8_ROMANIAN_CI

- UTF8_SLOVENIAN_CI
- UTF8_POLISH_CI
- UTF8_ESTONIAN_CI
- UTF8_SPANISH_CI
- UTF8_SWEDISH_CI
- UTF8_TURKISH_CI
- UTF8_CZECH_CI
- UTF8_DANISH_CI
- UTF8_LITHUANIAN_CI
- UTF8_SLOVAK_CI
- UTF8_SPANISH2_CI
- UTF8_ROMAN_CI
- UTF8_PERSIAN_CI
- UTF8_ESPERANTO_CI
- UTF8_HUNGARIAN_CI
- INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE.** Returns the name of the database specified in the authentication packet that the client sends to the database server. This is the `databasename` attribute.
- `MYSQL.CLIENT.USER.` Returns the user name (in the authentication packet) with which the client is attempting to connect to the database. This is the `user` attribute.
- **MYSQL.REQ.** Operates on a MySQL request.
- **MYSQL.REQ.COMMAND.** Identifies the enumeration constant assigned to the type of command in the request. The `EQ(<m>)` and `NE(<m>)` operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the enumeration constant values:
  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST

- CREATE_DB
- DROP_DB
- REFRESH
- SHUTDOWN
- STATISTICS
- PROCESS_INFO
- CONNECT
- PROCESS_KILL
- DEBUG
- PING
- TIME
- DELAYED_INSERT
- CHANGE_USER
- BINLOG_DUMP
- TABLE_DUMP
- CONNECT_OUT
- REGISTER_SLAVE
- STMT_PREPARE
- STMT_EXECUTE
- STMT_SEND_LONG_DATA
- STMT_CLOSE
- STMT_RESET
- SET_OPTION
- STMT_FETCH

- **MYSQL.REQ.QUERY.** Identifies the query in the MySQL request.
- **MYSQL.REQ.QUERY.COMMAND.** Returns the first keyword in the MySQL query.
- **MYSQL.REQ.QUERY.SIZE.** Returns the size of the request query in integer format. The SIZE method is similar to the CONTENT_LENGTH method that returns the length of an HTTP request or response.

- **MYSQL.REQ.QUERY.TEXT.** Returns a string covering the entire query.
- **MYSQL.REQ.QUERY.TEXT(<n>).** Returns the first `n` bytes of the MySQL query as a string. This is similar to `HTTP.BODY(<n>)`.

  Parameters:

  `n` - Number of bytes to be returned
- **MYSQL.RES.** Operates on a MySQL response.
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** Checks whether the response has at least `i` number of rows and returns a Boolean `TRUE` or `FALSE` to indicate the result.

  Parameters:

  `i` - Number of rows
- **MYSQL.RES.ERROR.** Identifies the MySQL error object. The error object includes the error number and the error message.
- **MYSQL.RES.ERROR.MESSAGE.** Returns the error message that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.NUM.** Returns the error number that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.SQLSTATE.** Returns the value of the `SQLSTATE` field in the server's error response. The MySQL server translates error number values to `SQLSTATE` values.
- **MYSQL.RES.FIELD(<i>).** Identifies the packet that corresponds to the i^{th} individual field in the server's response. Each field packet describes the properties of the associated column. The packet count (`i`) begins at 0.

  Parameters:

  `i` - Packet number
- **MYSQL.RES.FIELD(<i>).CATALOG.** Returns the `catalog` property of the field packet.
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** Returns the character set of the column. The `EQ(<m>)` and `NE(<m>)` operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.
- **MYSQL.RES.FIELD(<i>).DATATYPE.** Returns an enumeration constant that represents the data type of the column. This is the `type` (also called `enum_field_type`) attribute of the column. The `EQ(<m>)` and `NE(<m>)` operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. The possible values for the various data types are:
  - `DECIMAL`
  - `TINY`

- SHORT
- LONG
- FLOAT
- DOUBLE
- NULL
- TIMESTAMP
- LONGLONG
- INT24
- DATE
- TIME
- DATETIME
- YEAR
- NEWDATE
- VARCHAR (new in MySQL 5.0)
- BIT (new in MySQL 5.0)
- NEWDECIMAL (new in MySQL 5.0)
- ENUM
- SET
- TINY_BLOB
- MEDIUM_BLOB
- LONG_BLOB
- BLOB
- VAR_STRING
- STRING
- GEOMETRY

- **MYSQL.RES.FIELD(<i>).DB.** Returns the database identifier (db) attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).DECIMALS.** Returns the number of positions after the decimal point if the type is DECIMAL or NUMERIC. This is the decimals attribute of the field packet.

- **MYSQL.RES.FIELD(<i>).FLAGS.** Returns the `flags` property of the field packet. Following are the possible hexadecimal flag values:
  - `0001`: `NOT_NULL_FLAG`
  - `0002`: `PRI_KEY_FLAG`
  - `0004`: `UNIQUE_KEY_FLAG`
  - `0008`: `MULTIPLE_KEY_FLAG`
  - `0010`: `BLOB_FLAG`
  - `0020`: `UNSIGNED_FLAG`
  - `0040`: `ZEROFILL_FLAG`
  - `0080`: `BINARY_FLAG`
  - `0100`: `ENUM_FLAG`
  - `0200`: `AUTO_INCREMENT_FLAG`
  - `0400`: `TIMESTAMP_FLAG`
  - `0800`: `SET_FLAG`
- **MYSQL.RES.FIELD(<i>).LENGTH.** Returns the length of the column. This is the value of the `length` attribute of the field packet. The value that is returned might be larger than the actual value. For example, an instance of a `VARCHAR(2)` column might return a value of `2` even when it contains only one character.
- **MYSQL.RES.FIELD(<i>).NAME.** Returns the column identifier (the name after the `AS` clause, if any). This is the `name` attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** Returns the original column identifier (before the `AS` clause, if any). This is the `org_name` attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** Returns the original table identifier of the column (before the `AS` clause, if any). This is the `org_table` attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).TABLE.** Returns the table identifier of the column (after the `AS` clause, if any). This is the `table` attribute of the field packet.
- **MYSQL.RES.FIELDS_COUNT.** Returns the number of field packets in the response (the `field_count` attribute of the `OK` packet).
- **MYSQL.RES.OK.** Identifies the `OK` packet sent by the database server.
- **MYSQL.RES.OK.AFFECTED_ROWS.** Returns the number of rows affected by an `INSERT`, `UPDATE`, or `DELETE` query. This is the value of the `affected_rows` attribute of the `OK` packet.
- **MYSQL.RES.OK.INSERT_ID.** Identifies the `unique_id` attribute of the `OK` packet. If an auto-increment identity is not generated by the current MySQL statement or query, the value of `unique_id`, and hence the value returned by the expression, is `0`.

- **MYSQL.RES.OK.MESSAGE.** Returns the `message` property of the `OK` packet.
- **MYSQL.RES.OK.STATUS.** Identifies the bit string in the `server_status` attribute of the `OK` packet. Clients can use the server status to check whether the current command is a part of a running transaction. The bits in the `server_status` bit string correspond to the following fields (in the given order):
  - `IN TRANSACTION`
  - `AUTO_COMMIT`
  - `MORE RESULTS`
  - `MULTI QUERY`
  - `BAD INDEX USED`
  - `NO INDEX USED`
  - `CURSOR EXISTS`
  - `LAST ROW SEEN`
  - `DATABASE DROPPED`
  - `NO BACKSLASH ESCAPES`
- **MYSQL.RES.OK.WARNING_COUNT.** Returns the `warning_count` attribute of the `OK` packet.
- **MYSQL.RES.ROW(<i>).** Identifies the packet that corresponds to the i^{th} individual row in the database server's response.

  Parameters:

  `i` - Row number
- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** Checks whether the j^{th} column of the i^{th} row of the table is `NULL`. Following C conventions, both indexes `i` and `j` start from 0. Therefore, row `i` and column `j` are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.

  Parameters:

  `i` - Row number

  `j` - Column number
- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** Checks whether the j^{th} column of the i^{th} row of the table is `NULL`. Following C conventions, both indexes `i` and `j` start from 0. Therefore, row `i` and column `j` are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.

  Parameters:

  `i` - Row number

`j` - Column number

- **`MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).`** Returns an integer value from the j^{th} column of the i^{th} row of the table. Following C conventions, both indexes `i` and `j` start from 0. Therefore, row `i` and column `j` are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.

  Parameters:

  `i` - Row number

  `j` - Column number

- **`MYSQL.RES.ROW(<i>).TEXT_ELEM(j).`** Returns a string from the j^{th} column of the i^{th} row of the table. Following C conventions, both indexes `i` and `j` start from 0. Therefore, row `i` and column `j` are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.

  Parameters:

  `i` - Row number

  `j` - Column number

- **`MYSQL.RES.TYPE.`** Returns an enumeration constant for the response type. Its values can be `ERROR`, `OK`, and `RESULT_SET`. The `EQ(<m>)` and `NE(<m>)` operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.

Expressions for Evaluating Microsoft SQL Server Connections

The following expressions evaluate traffic associated with Microsoft SQL Server database servers. You can use the request-based expressions (expressions that begin with MSSQL.CLIENT and MSSQL.REQ) in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MSSQL.RES) to evaluate server responses to user-configured health monitors.

Table 1. Expressions for Evaluating Microsoft SQL Server Connections

Expression	Description
MSSQL.CLIENT.CAPABILITIES	Returns the OptionFlags1, OptionFlags2, OptionFlags3, and TypeFlags fields of the LOGIN7 authentication packet, in that order, as a 4-byte integer. Each field is 1 byte long and specifies a set of client capabilities.
MSSQL.CLIENT.DATABASE	Returns the name of the client database. The value returned is of type text.
MSSQL.CLIENT.USER	Returns the user name with which the client authenticated. The value returned is of type text.
MSSQL.REQ.COMMAND	Returns an enumeration constant that identifies the type of command in the request sent to a Microsoft SQL Server database server. The value returned is of type text. Examples of the values of the enumeration constant are QUERY, RESPONSE, RPC, and ATTENTION. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.
MSSQL.REQ.QUERY.COMMAND	Returns the first keyword in the SQL query. The value returned is of type text.
MSSQL.REQ.QUERY.SIZE	Returns the size of the SQL query in the request. The value returned is a number.
MSSQL.REQ.QUERY.TEXT	Returns the entire SQL query as a string. The value returned is of type text.
MSSQL.REQ.QUERY.TEXT(<n>)	Returns the first n bytes of the SQL query. The value returned is of type text. **Parameters:** n - Number of bytes
MSSQL.REQ.RPC.NAME	Returns the name of the procedure that is being called in a remote procedure call (RPC) request. The name is returned as a string.

MSSQL.REQ.RPC.IS_PROCID	Returns a Boolean value that indicates whether the remote procedure call (RPC) request contains a procedure ID or an RPC name. A return value of TRUE indicates that the request contains a procedure ID and a return value of FALSE indicates that the request contains an RPC name.
MSSQL.REQ.RPC.PROCID	Returns the procedure ID of the remote procedure call (RPC) request as an integer.
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	Checks whether the response has at least i number of rows. The value returned is a Boolean TRUE or FALSE value. **Parameters:** i - Number of rows
MSSQL.RES.DONE.ROWCOUNT	Returns a count of the number of rows affected by an INSERT, UPDATE, or DELETE query. The value returned is of type unsigned long.
MSSQL.RES.DONE.STATUS	Returns the status field from the DONE token sent by a Microsoft SQL Server database server. The value returned is a number.
MSSQL.RES.ERROR.MESSAGE	Returns the error message from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the MsgText field in the ERROR token. The value returned is of type text.
MSSQL.RES.ERROR.NUM	Returns the error number from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the Number field in the ERROR token. The value returned is a number.
MSSQL.RES.ERROR.STATE	Returns the error state from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the State field in the ERROR token. The value returned is a number.
MSSQL.RES.FIELD(<i>).DATATYPE	Returns the data type of the i^{th} field in the server response. The EQ(<m>) and NE(<m>) functions, which return Boolean values to indicate the result of a comparison, are used with this prefix. For example, the following expression returns a Boolean TRUE if the DATATYPE function returns a value of datetime for the third field in the response: MSSQL.RES.FIELD(<2>).DATATYPE.EQ(datetime) **Parameters:** i - Row number
MSSQL.RES.FIELD(<i>).LENGTH	Returns the maximum possible length of the i^{th} field in the server response. The value returned is a number. **Parameters:** i - Row number

MSSQL.RES.FIELD(<i>).NAME	Returns the name of the i^{th} field in the server response. The value returned is of type text. **Parameters:** i - Row number
MSSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>)	Returns a value of type double from the j^{th} column of the i^{th} row of the table. If the value is not a double value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number
MSSQL.RES.ROW(<i>).NUM_ELEM(j)	Returns an integer value from the j^{th} column of i^{th} row of the table. If the value is not an integer value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number
MSSQL.RES.ROW(<i>).IS_NULL_ELEM(j)	Checks whether the j^{th} column of the i^{th} row of the table is NULL and returns a Boolean TRUE or FALSE to indicate the result. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number
MSSQL.RES.ROW(<i>).TEXT_ELEM(j)	Returns a text string from the j^{th} column of i^{th} row of the table. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number

MSSQL.RES.TYPE	Returns an enumeration constant that identifies the response type. Following are the possible return values: • ERROR • OK • RESULT_SET The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.

Typecasting Data

You can extract data of one type (for example, text or an integer) from requests and responses and transform it to data of another type. For example, you can extract a string and transform the string to time format. You can also extract a string from an HTTP request body and treat it like an HTTP header or extract a value from one type of request header and insert it in a response header of a different type.

After typecasting the data, you can apply any operation that is appropriate for the new data type. For example, if you typecast text to an HTTP header, you can apply any operation that is applicable to HTTP headers to the returned value.

The following table describes various typecasting operations.

Table 1. Typecasting Functions

	Description

TYPECAST_LIST_T(<separator>)	Treats the text in an HTTP request or response body as a list whose elem by the character in the <separator> argument. Index values in the list th with zero (0). Text mode settings have no effect on the separator. For example, even i mode to IGNORECASE, and the separator is the letter “p,” an uppercase a separator. The following example creates a Rewrite action that constructs a list fro body and extracts the fourth item in the list: add rewrite action myreplace_action REPLACE 'http.req.body(100)' 'http.req.body(100).typecast_list_t('?').get(4)' set rewrite policy myreplace_policy -action myreplace_action This policy returns the string “fourth item” from the following request: GET?first item?second item?third item?fourth item? The following example extracts the fourth-from-last item from the list. add rewrite action myreplace_action1 REPLACE 'http.req.body(100)' 'http.req.body(100).typecast_list_t('?').get_reverse(4)' set rewrite policy myreplace_policy1 -action myreplace_action1 This policy returns the string “first item” from the following request: GET?first item?second item?third item?fourth item.

TYPECAST_ NVLIST_T(<separator>, <delimiter>) PECAST_ NVLIST_T(<separator>, <delimiter>,)	Treats the text as a name-value list. The <separator> argument identifie separates the name and the value. The <delimiter> argument identifies separates each name-value pair. The <quote> character is required whe into a name-value list that supports quoted strings. Any delimiters that a quoted string are ignored. The text mode has no effect on the delimiters. For example, if the curre IGNORECASE and you specify “p” as the delimiter, an uppercase “P” is n delimiter. For example, the following policy counts the number of name-value pair result in a header named name-value-count: add rewrite action mycount_action insert_http_header name-value-coun 'http.req.header("Cookie").typecast_nvlist_t('=',';').count' set rewrite policy mycount_policy -action mycount_action This policy can extract a count of arguments in Cookie headers and inser name-value-count header: `Cookie: name=name1; rank=rank1`
TYPECAST_TIME_T	Treats the designated text as a date string. The following formats are su • RFC822: Sun, 06 Nov 1994 08:49:37 GMT • RFC850: Sunday, 06-Nov-94 08:49:37 GMT • ASCII TIME: Sun Nov 6 08:49:37 1994 • HTTP Set-Cookie Expiry date: Sun, 06-Nov-1994 08:49:37 GMT For example, the following policy converts the string to a time value and day. This policy matches all requests that have a day value lesser than o Add rewrite policy mytime_policy "http.req.body(100) .typecast_time_t.day.le(10)" mytime_action bind rewrite global mytime_policy 100

c string>.TYPECAST_IP_ ADDRESS_T	Treats a numeric string as an IP address. For example, the following policy matches HTTP requests that contains C a value of: 12.34.56.78\r\n. set rewrite policy ip_check_policy -rule 'http.req.cookie .value("ip").typecast_ip_address_t.eq(12.34.56.78)' bind rewrite global ip_check_policy 200 -type req_default
c string>.TYPECAST_IPV6_ADDRESS_T	Treats a string as an IPv6 address in the following format: 0000:0000:CD00:0000:0000:00AB:0000:CDEF
TYPECAST_HTTP_ URL_T	Treats the designated text as the URL in the first line of an HTTP reques supported format is [<protocol>://<hostname>]<path>?<query> is set to URLENCODED by default. For example, the following policy replaces a URL-encoded part of a strin named Test. add rewrite action replace_header_string replace "http.req.header(\"Test\").typecast_http_url_t.path .before_str(\"123\").after_str(\"ABC\")" "\"string\"" add rewrite policy rewrite_test_header_policy true replace_header_strin bind rewrite global rewrite_test_header_policy 1 END -type res_override Consider the following header: Test: ABC%12123\r\n This policy would replace the preceding header with the value ABC%str
TYPECAST_HTTP_ HOSTNAME_T	Provides operations for parsing an HTTP host name as it appears in HTTP for a host name is abc.def.com:8080.
TYPECAST_HTTP_ METHOD_T	Converts text to an HTTP method. For example, the following policy matches any HTTP request that contai with a value equal to POST: Add rewrite policy method_policy "http.req.header(\"Host\") .typecast_http_method_t.eq(POST)" act1
TYPECAST_DNS_ DOMAIN_T	Enables the designated text to be parsed like a DNS domain name in the

TYPECAST_HTTP_ HEADER_T("<name>")	Converts the designated text to a multi-line HTTP header that you specif argument. For example, the following expression converts “MyHeader” to “InHeade http.req.header("MyHeader").typcast_http_header_t("InHeader") Typically, text operations that you specify in this type of expression app line of this header, with some exceptions. For example, the CONTAINS o values in all the lines in instances of this header type.
TYPECAST_COOKIE_T	Treats the designated text as an HTTP cookie as it appears in a Set-Cook header. You can apply name-value list operations as well as text operati designated text. For example, you can designate equals (=) as the name- the semicolon (;) as the list element delimiter. If you apply name-value list operations, the list is parsed as if IGNORE_E were in effect. Each cookie begins with a `cookie-name=cookie-value` pair, optiona attribute-value pairs that are separated by a semicolon, as follows: `cookie1=value1;version=n.n;value;domain=value;path=valu` If the same attribute appears more than once in a cookie, the value for t the attribute is returned.
>.TYPECAST_DOUBLE_AT	Transforms the number to a value of data type double.
>.TYPECAST_IP_ADDRESS_AT	Converts the number to an IP address.
>.TYPECAST_TIME_AT	Converts the number to time format.

>.TYPECAST_TIME_AT.BETWEEN(<time1>, <time2>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the tim by <number> is between the lower and upper time value arguments <tim The following are prerequisites for this function: • Both the lower and upper time arguments must be fully specified. Fo 1995 Jan is fully specified. But GMT Jan, GMT 1995 20 and GMT Jan / specified. • Both arguments must be either GMT or Local. • The day of the week must not be present in either argument. Howev month can be specified as the first, second, third, or fourth weekday (example Wed_3 is the third Wednesday of the month). • The upper time argument, <time2>, must be bigger than the lower t <time1>. The following examples assume that the current time value is GMT 2005 and that the day is the first Sunday of the month of May in 2005. The res is given after each example. BETWEEN(GMT 2004, GMT 2006): TRUE BETWEEN(GMT 2004 Jan, GMT 2006 Nov): TRUE BETWEEN(GMT 2004 Jan, GMT 2006): TRUE BETWEEN(GMT 2005 May Sun_1, GMT 2005 May Sun_3): TRUE BETWEEN(GMT 2005 May 1, GMT May 2005 1): TRUE BETWEEN(LOCAL 2005 May 1, LOCAL May 2005 1): The result depends on the NetScaler system's timezone. Parameters: <time1> - Lower time value <time2> - Upper time value
>.TYPECAST_TIME_AT.DAY	Extracts the day of the month from the current system time and returns number that corresponds to the day of the month. The returned value ra

>.TYPECAST_TIME_AT.EQ(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the tim by <number> is equal to the time value argument <t>. The following examples assume that the current time value is GMT 2005 and that the day is the 1st Sunday of the month of May in 2005. The resu is given after each example. EQ(GMT 2005): TRUE EQ(GMT 2005 Dec): FALSE EQ(Local 2005 May): TRUE or FALSE, depending on the time zone. EQ(GMT 10h): TRUE EQ(GMT 10h 30s): TRUE EQ(GMT May 10h): TRUE EQ(GMT Sun): TRUE EQ(GMT May Sun_1): TRUE Parameters: <t> - Time
>.TYPECAST_TIME_AT.GE(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the tim by <number> is greater than or equal to the time value argument <t>. The following examples assume that the current time value is GMT 2005 and that the day is the 1st Sunday of the month of May in 2005. The resu is given after each example. GE(GMT 2004): TRUE GE(GMT 2005 Jan): TRUE GE(Local 2005 May): TRUE or FALSE, depending on the time zone. GE(GMT 8h): TRUE GE(GMT 30m): FALSE GE(GMT May 10h): TRUE GE(GMT May 10h 0m): TRUE GE(GMT Sun): TRUE GE(GMT May Sun_1): TRUE Parameters: <t> - Time

>.TYPECAST_TIME_AT.GT(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the tim by <number> is greater than the time value argument <t>. The following examples assume that the current time value is GMT 2005 and that the day is the 1st Sunday of the month of May in 2005. The resu is given after each example. GT(GMT 2004): TRUE GT(GMT 2005 Jan): TRUE GT(Local 2005 May): TRUE or FALSE, depending on the time zone. GT(GMT 8h): TRUE GT(GMT 30m): FALSE GT(GMT May 10h): FALSE GT(GMT May 10h 0m): TRUE GT(GMT Sun): FALSE GT(GMT May Sun_1): FALSE Parameters: <t> - Time
>.TYPECAST_TIME_AT.HOURS	Extracts the hour from the current system time and returns the correspo integer that can range from 0 to 23.
>.TYPECAST_TIME_AT.LE(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the tim by <number> is lesser than or equal to the time value argument <t>. The following examples assume that the current time value is GMT 2005 and that the day is the 1st Sunday of the month of May in 2005. The resu is given after each example. LE(GMT 2006): TRUE LE(GMT 2005 Dec): TRUE LE(Local 2005 May): TRUE or FALSE, depending on the time zone. LE(GMT 8h): FALSE LE(GMT 30m): TRUE LE(GMT May 10h): TRUE LE(GMT Jun 11h): TRUE LE(GMT Wed): TRUE LE(GMT May Sun_1): TRUE Parameters: <t> - Time

>.TYPECAST_TIME_AT.LT(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time by <number> is lesser than the time value argument <t>. The following examples assume that the current time value is GMT 2005 and that the day is the 1st Sunday of the month of May in 2005. The resu is given after each example. LT(GMT 2006): TRUE LT(GMT 2005 Dec): TRUE LT(Local 2005 May): TRUE or FALSE, depending on the time zone. LT(GMT 8h): FALSE LT(GMT 30m): TRUE LT(GMT May 10h): FALSE LT(GMT Jun 11h): TRUE LT(GMT Wed): TRUE LT(GMT May Sun_1): FALSE Parameters: <t> - Time
>.TYPECAST_TIME_AT.MINUTES	Extracts the minute from the current system time and returns the value can range from 0 to 59.
>.TYPECAST_TIME_AT.MONTH	Extracts the month from the current system time and returns the value a can range from 1 (January) to 12 (December).
>.TYPECAST_TIME_AT.RELATIVE_BOOT	Calculates the number of seconds that have elapsed after the most recen number of seconds to the next scheduled reboot, depending on which is time, and returns an integer. If the closest boot time is in the past, the If the closest boot time is in the future (scheduled reboot time), the inte
>.TYPECAST_TIME_AT.RELATIVE_NOW	Calculates the number of seconds between the current system time and and returns the value as an integer. If the designated time is in the past negative. If it is in the future, the integer is positive.
>.TYPECAST_TIME_AT.SECONDS	Extracts the seconds from the current system time and returns the value can range from 0 to 59.
>.TYPECAST_TIME_AT.WEEKDAY	Returns an integer that corresponds to the day of the week; 0 for Sunday

>.TYPECAST_TIME_AT.WITHIN(<time1>, <time2>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the tim by <number> lies within all the ranges defined by lower and upper time <time1> and <time2>. If an element of time such as the day or the hour is left unspecified in th <time1>, then it is assumed to have the lowest value possible for its rang If an element is left unspecified in the upper argument , <time2>, then i the highest value possilbe for its range. If the year is specified in one of the arguments, then it must be specifie argument as well. Following are the ranges for different elements of time: • month: 1-12 • day: 1-31 • weekday: 0-6 • hour: 0-23 • minutes: 0-59 • seconds: 0-59. Each element of time in the lower time value argument defines a range the corresponding element in the upper time value argument. For the re each element of time in the time value designated by <number> must lie corresponding range specified by the lower and upper arguments. The following examples assume that the current time value is GMT 2005 30s.and that the day is the second Tuesday of the month. The result of t given after each example. WITHIN(GMT 2004, GMT 2006): TRUE WITHIN(GMT 2004 Jan, GMT 2006 Mar): FALSE (May doesn't fall in the Jan WITHIN(GMT Feb, GMT): TRUE (May falls in the Feb-Dec range.) WITHIN(GMT Sun_1, GMT Sun_3): TRUE (2nd Tuesday lies within 1st Sund and the 3rd Sunday.) WITHIN(GMT 2005 May 1 10h, GMT May 2005 1 17h): TRUE WITHIN(LOCAL 2005 May 1, LOCAL May 2005 1): The result depends on the NetScaler system's timezone. Parameters: <time1> - Lower time value <time2> - Upper time value

>.TYPECAST_TIME_AT.YEAR	Extracts the year from the current system time and returns the value as
>.TYPECAST_NUM_T(<type>)	Casts numeric string data to a signed 32-bit number. The argument <typ the following: • DECIMAL. Treat the string as a decimal number and cast to a signed • HEX. Treat the string as a hexadecimal number and cast to a signed • DECIMAL_PREFIX. Consider the part of the string up to the first occ character that is not a valid decimal character and cast to a signed 3 • HEX_PREFIX. Consider the part of the string up to the first occurrer that is not a valid hexadecimal character and cast to a signed 32-bit For example, the following policy extracts a numeric portion of a query s number, and inserts an HTTP header named Company with the resulting add rewrite action myadd_action insert_http_header Company "http.req.url.query.typecast_num_t(decimal).add(4)" add rewrite policy myadd_policy true myadd_action bind rewrite global myadd_policy 300 END -type RES_DEFAULT For example, this policy would extract “4444” from the following URL st /test/file.html?4444 The action that is associated with the policy would insert the following H header: Company: 4448\r\n
>.TYPECAST_NUM_AT	Casts a number of any data type to a number of data type integer.
>.TYPECAST_DOUBLE_AT	Casts a number of any data type to a number of data type double.
>.TYPECAST_UNSIGNED_LONG_AT	Casts a number of any data type to a number of data type unsigned long
>.TYPECAST_NUM_T(<type>,<default>)	Casts string data to a signed 32-bit number. If the typecasting operation (UNDEF) condition, the function returns the value specified for default. takes the values specified for TYPECAST_NUM_T(<type>).
>.TYPECAST_UNSIGNED_LONG_T(<type>)	Casts string data to data of type unsigned long. The argument can be on • DECIMAL. Treat the string as a decimal number and cast to unsigned • HEX. Treat the string as a hexadecimal number and cast to unsigned • DECIMAL_PREFIX. Consider the part of the string up to the first occ character that is not a valid decimal character and cast to unsigned • HEX_PREFIX. Consider the part of the string up to the first occurrer that is not a valid hexadecimal character and cast to unsigned long.

>.TYPECAST_UNSIGNED_LONG_T(<type>,<default>)	Casts string data to data of type unsigned long. If the typecasting operat undefined (UNDEF) condition, the function returns the value specified fo argument takes the values specified for TYPECAST_UNSIGNED_LONG_T

Regular Expressions

When you want to perform string matching operations that are more complex than the operations that you perform with the CONTAINS(“<string>”) or EQ(“<string>“) operators, you use regular expressions. The policy infrastructure on the Citrix® NetScaler® appliance includes operators to which you can pass regular expressions as arguments for text matching. The names of the operators that work with regular expressions include the string REGEX. The regular expressions that you pass as arguments must conform to the regular expression syntax that is described in http://www.pcre.org/pcre.txt. You can learn more about regular expressions at http://www.regular-expressions.info/quickstart.htmland at http://www.silverstones.com/thebat/Regex.html.

The target text for an operator that works with regular expressions can be either text or the value of an HTTP header. Following is the format of a default syntax expression that uses a regular expression operator to operate on text:

```
<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)
```

The string <text> represents the default syntax expression prefix that identifies a text string in a packet (for example, HTTP.REQ.URL). The string <regex_operator> represents the regular expression operator. The regular expression always begins with the string re. A pair of matching delimiters, represented by <delimiter>, enclose the string <regex_pattern>, which represents the regular expression.

The following example expression checks whether the URL in an HTTP packet contains the string *.jpeg (where * is a wildcard) and returns a Boolean TRUE or FALSE to indicate the result. The regular expression is enclosed within a pair of slash marks (/), which act as delimiters.

```
http.req.url.regex_match(re/*.jpeg/)
```

Regular expression operators can be combined to define or refine the scope of a search. For example,
`<text>.AFTER_REGEX(re/regex_pattern1/).BEFORE_REGEX(re/regex_pattern2/)`
specifies that the target for string matching is the text between the patterns regex_pattern1 and regex_pattern2. You can use a text operator on the scope that is defined by the regular expression operators. For example, you can use the CONTAINS(“<string>”) operator to check whether the defined scope contains the string abc:

```
<text>.AFTER_REGEX(re/regex_pattern1/).BEFORE_REGEX(re/regex_pattern2/).CONTAINS(
```

Note: The process of evaluating a regular expression inherently takes more time than that for an operator such as CONTAINS(“<string>”) or EQ(“<string>”), which work with simple string arguments. You should use regular expressions only if your requirement is beyond the scope of other operators.

Basic Characteristics of Regular Expressions

Following are notable characteristics of regular expressions as defined on the NetScaler appliance:

- A regular expression always begins with the string “re” followed by a pair of delimiting characters (called delimiters) that enclose the regular expression that you want to use.

  For example, `re#<regex_pattern>#` uses the number sign (#) as a delimiter.

- A regular expression cannot exceed 1499 characters.
- Digit matching can be done by using the string \d (a backslash followed by d).
- White space can be represented by using \s (a backslash followed by s).
- A regular expression can contain white spaces.

Following are the differences between the NetScaler syntax and the PCRE syntax:

- The NetScaler does not allow back references in regular expressions.
- You should not use recursive regular expressions.
- The dot meta-character also matches the newline character.
- Unicode is not supported.
- The operation SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option in the regular expression.

Operations for Regular Expressions

The following table describes the operators that work with regular expressions. The operation performed by a regular expression operator in a given default syntax expression depends on whether the expression prefix identifies text or HTTP headers. Operations that evaluate headers override any text-based operations for all instances of the specified header type. When you use an operator, replace <text> with the default syntax expression prefix that you want to configure for identifying text.

Table 1. Default Syntax Expression Operators That Work with Regular Expressions

sion Operation	Description
RE_REGEX(<regular)	Selects the text that precedes the string that matches the <regular expression> argument. If the regular exp any data in the target, the expression returns a text object of length 0. The following expression selects the string "text" from "text/plain". `http.res.header("content-type").before_regex(re#/#)`
R_REGEX(<regular)	Selects the text that follows the string that matches the <regular expression> argument. If the regular expre text in the target, the expression returns a text object of length 0. The following expression extracts "Example" from "myExample": `http.req.header("etag").after_regex(re/my/)`
X_SELECT(<regular)	Selects a string that matches the <regular expression> argument. If the regular expression does not match t of length 0 is returned. The following example extracts the string "NS-CACHE-9.0: 90" from a Via header: `http.req.header("via").regex_select(re!NS-CACHE-\d\.\d:\s*\d{1,3}!)`

`X_MATCH(<regular )`	Returns TRUE if the target matches a <regular expression> argument of up to 1499 characters.
	
	The regular expression must be of the following format:
	
	`re<delimiter>regular expression< delimiter>`
	
	Both delimiters must be the same. Additionally, the regular expression must conform to the Perl-compatible expression library syntax. For more information, go tohttp://www.pcre.org/pcre.txt. In particular, see the However, note the following:
	
	• Back-references are not allowed.
	• Recursive regular expressions are not recommended.
	• The dot metacharacter also matches the newline character.
	• The Unicode character set is not supported.
	• SET_TEXT_MODE(IGNORECASE) overrides the “(?i)” internal option specified in the regular expression.
	
	The following are examples:
	
	http.req.hostname.regex_match(re/[[:alpha:]]+(abc){2,3}/)
	http.req.url.set_text_mode(urlencoded).regex_match(re#(a*b+c*)#)
	
	The following example matches ab and aB:
	
	`http.req.url.regex_match(re/a(?i)b/)`
	
	The following example matches ab, aB, Ab and AB:
	
	`http.req.url.set_text_mode(ignorecase).regex_match(re/ab/)`
	
	The following example performs a case-insensitive, multiline match in which the dot meta-character also m character:
	
	`http.req.body.regex_match(re/(?ixm) (^ab (.*) cd$) /)`

Configuring Classic Policies and Expressions

Some NetScaler features use classic policies and classic expressions. As with default syntax policies, classic policies can be either global or specific to a virtual server. However, to a certain extent, the configuration method and bind points for classic policies are different from those of default syntax policies. As with default syntax expressions, you can configure named expressions and use a named expression in multiple classic policies.

Where Classic Policies Are Used

The following table summarizes NetScaler features that can be configured by using classic policies.

Table 1. Policy Type and Bind Points for Policies in Features That Use Classic Policies

ature	Virtual Servers	Supported Policies	Policy Bind Points	How Polici
tem features, thentication	None	Authentication policies	Global	For th Authe featu conta authe schen differ authe meth exam confi certif authe schen
	None	SSL policies	• Global • Load Balancing virtual server	To de to ap encry and a inform clear To pr end-t secur messa decry featu clear SSL to comm back- serve

ntent itching an use either ssic or default ntax policies, t not both)	Content Switching virtual server	Content Switching policies	• Content Switching virtual server • Cache Redirection virtual server	To de serve serve respo servi based chara an in reque Requ chara inclu type, cooki meth type assoc serve
mpression	None	HTTP Compression policies	• Global • Content Switching virtual server • Load Balancing virtual server • SSL Offload virtual server • Service	To de type traffi comp
otection tures, Filter	None	Content Filtering policies	• Global • Content Switching virtual server • Load Balancing virtual server • SSL Offload virtual server • Service	To co behav filter
otection tures, eConnect	None	SureConnect policies	• Load Balancing virtual server • SSL Offload virtual server • Service	To co behav SureC funct
otection tures, Priority eueing	None	Priority Queueing policies	• Load Balancing virtual server • SSL Offload virtual server	To co behav Priori funct

ML Injection	None	HTML Injection Policies	• Global • Load Balancing virtual server • Content Switching virtual server • SSL Offload virtual server	To en NetSc text c an HT that i client
A - Traffic nagement	None	Authentication, Authorization, Auditing, and Session policies	• Authentication virtual server (authentication, session, and auditing policies) • Load Balancing or Content Switching virtual server (authorization and auditing policies) • Global (session and audit policies) • AAA group or user (session, auditing, and authorization policies)	To co for us speci and a user a
che direction	Cache Redirection virtual server	Cache Redirection policies Map policies	Cache Redirection virtual server	To de whetl respo serve cache serve
olication ewall	None	Application Firewall policies	Global	To ide chara traffi that s shoul admit the fi

cess Gateway	VPN server	Pre-Authentication policies	• AAA Global • VPN vserver	To de the A Gatev authe autho audit funct defin rules Web the A Gatev
		Authentication policies	• System Global • AAA Global • VPN vserver	
		Auditing policies	• User • User group • VPN vserver	
		Session policies	• VPN Global • User • User Group • VPN vserver	
		Authorization policies	• User • User Group	
		Traffic policies	• VPN Global • User • User Group • VPN vserver	
		TCP Compression policies	VPN Global	

Configuring a Classic Policy

You can configure classic policies and classic expressions by using either the configuration utility or the command-line interface. A policy rule cannot exceed 1,499 characters. When configuring the policy rule, you can use named classic expressions. For more information about named expressions, see Creating Named Classic Expressions. After configuring the policy, you bind it either globally or to a virtual server.

Note that there are small variations in the policy configuration methods for various NetScaler features.

Note: You can embed a classic expression in a default syntax expression by using the syntax SYS.EVAL_CLASSIC_EXPR***(classic_expression)***, specifying the *classic_expression* as the argument.

To create a classic policy by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add <featureName> policy <name> **-rule** <expression> **-action** <action>
- show <featureName> policy [<policyName>]

Example

The following commands first create a compression action and then create a compression policy that applies the action:

```
> add cmp action cmp-act-compress compress
 Done
> show cmp action cmp-act-compress
1)      Name: cmp-act-compress  Compression Type: compress
 Done
> add cmp pol cmp-pol-compress -rule ExpCheckIp -resAction cmp-act-compress
 Done
> show cmp pol cmp-pol-compress
1)      Name: cmp-pol-compress  Rule: ExpCheckIp
       Response action: cmp-act-compress       Hits: 0
 Done
>
```

Parameters for configuring a classic policy

featureName

The feature for which you are creating the policy. For example, for Access Gateway policies, type `accessgw`. For Application Firewall policies, type `appfw`. For SSL policies, type `ssl`.

name

A name for the policy. You must begin a policy name with a letter or underscore. A policy name can consist of 1 to 31 characters, including letters, numbers, hyphen (-), period (.), pound sign (#), space ( ), and underscore (_).

expression

The expression, as described in Configuring a Classic Expression.

action

The name of the action that you want to associate with this policy. For Access Gateway and Application Firewall policies, you substitute the appropriate profile instead of an action.

To create a policy with classic expressions by using the configuration utility

1. In the navigation pane, expand the feature for which you want to configure a policy and, depending on the feature, do the following:
   - For Content Switching, Cache Redirection, and the Application Firewall, click **Policies**.
   - For SSL, click **Policies**, and then in the details pane, click the **Policies** tab.
   - For System Authentication, click **Authentication**, and then in the details pane, click the **Policies** tab.
   - For Filter, SureConnect, and Priority Queuing, expand **Protection Features**, select the desired function, and then in the details pane, click the **Policies** tab.
   - For the Access Gateway, expand **Access Gateway**, expand **Policies**, select the desired function, and then in the details pane, click the **Policies** tab.
2. For most features, click the **Add** button.
3. In the **Create <feature name> Policy** dialog box, in the **Name*** text box, enter a name for the policy.

   Note: Note: You must begin a policy name with a letter or underscore. A policy name can consist of 1 to 31 characters, including letters, numbers, hyphen (-), period (.), pound sign (#), space ( ), and underscore (_).
4. For most features, you associate an action or a profile. For example, you may be required to select an action, or, in the case of an Access Gateway or Application Firewall policy, you select a profile to associate with the policy. A profile is a set of configuration options that operate as a set of actions that are applied when the data being analyzed matches the policy rule.

   For more information about creating a profile, seeConfiguring Policies and Profiles on the Access Gateway.
5. Create an expression that describes the type of data that you want this policy to match.

   Depending on the type of policy you want to create, you can choose a predefined expression, or you can create a new expression. For instructions on how to create an expression for most types of classic policies, see Configuring a Classic Expression.

   Named expressions are predefined expressions that you can reference by name in a policy rule. For more information about named expressions, see Creating Named Classic Expressions. For a list of all the default named expressions and a definition of each, see Expressions Reference.
6. Click **Create** to create your new policy.
7. Click **Close** to return to the Policies screen for the type of policy you were creating.

Configuring a Classic Expression

Classic expressions consist of the following expression elements, listed in hierarchical order:

- **Flow Type.** Specifies whether the connection is incoming or outgoing. The flow type is REQ for incoming connections and RES for outgoing connections.
- **Protocol.** Specifies the protocol, the choices for which are HTTP, SSL, TCP, and IP.
- **Qualifier.** The protocol attribute, which depends on the selected protocol.
- **Operator.** The type of test you want to perform on the connection data. Your choice of operator depends upon the connection information you are testing. If the connection information you are testing is text, you use text operators. If it is a number, you use standard numeric operators.
- **Value.** The string or number against which the connection data element—defined by the flow type, protocol, and qualifier—is tested. The value can be either a literal or an expression. The literal or expression must match the data type of the connection data element.

In a policy, classic expressions can be combined to create more complex expressions using Boolean and comparative operators.

Expression elements are parsed from left to right. The leftmost element is either REQ or RES and designates a request or a response, respectively. Successive terms define a specific connection type and a specific attribute for that connection type. Each term is separated from any preceding or following term by a period. Arguments appear in parentheses and follow the expression element to which they are passed.

The following classic expression fragment returns the client source IP for an incoming connection.

```
REQ.IP.SOURCEIP
```

The example identifies an IP address in a request. The expression element SOURCEIP designates the source IP address. This expression fragment may not be useful by itself. You can use an additional expression element, an operator, to determine whether the returned value meets specific criteria.The following expression tests whether the client IP is in the subnet 200.0.0.0/8 and returns a Boolean TRUE or FALSE:

```
REQ.IP.SOURCEIP == 200.0.0.0 -netmask 255.0.0.0
```

To create a classic policy expression by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- set <featureName> policy <name> **-rule** <expression> **-action** <action>
- show <featureName> policy <name>

Example

```
> set appfw policy GenericApplicationSSL_ 'HTTP.REQ.METHOD.EQ("get")' APPFW_DROP
 Done
> show appfw policy GenericApplicationSSL_
      Name: GenericApplicationSSL_    Rule: HTTP.REQ.METHOD.EQ("get")
      Profile: APPFW_DROP     Hits: 0
      Undef Hits: 0
      Policy is bound to following entities
      1) REQ VSERVER app_u_GenericApplicationSSLPortalPages   PRIORITY : 100
 Done
```

To add an expression for a classic policy by using the configuration utility

This procedure documents the **Add Expression** dialog box. Depending on the feature for which you are configuring a policy, the route by which you arrive at this dialog box may be different.

1. Perform steps 1-4 in To create a policy with classic expressions by using the configuration utility.
2. In the **Add Expression** dialog box, in **Expression Type**, click the type of expression you want to create.
3. Under **Flow Type**, click the down arrow and choose a flow type.

   The flow type is typically REQ or RES. The REQ option specifies that the policy applies to all incoming connections or requests. The RES option applies the policy to all outgoing connections or responses.

   For Application Firewall policies, you should leave the expression type set to **General Expression**, and the flow type set to **REQ**. The Application Firewall treats each request and response as a single paired entity, so all Application Firewall policies begin with **REQ**.

4. Under **Protocol**, click the down arrow and choose the protocol you want for your policy expression. Your choices are:

   - **HTTP.** Evaluates HTTP requests that are sent to a Web server. For classic expressions, HTTP includes HTTPS requests.
   - **SSL.** Evaluates SSL data associated with the current connection.
   - **TCP.** Evaluates the TCP data associated with the current connection.
   - **IP.** Evaluates the IP addresses associated with the current connection.

5. Under **Qualifier**, click the down arrow and choose a qualifier for your policy.

   The qualifier defines the type of data to be evaluated. The list of qualifiers that appears depends on which protocol you selected in step 4.

   The following list describes the qualifier choices for the HTTP protocol. For a complete list of protocols and qualifiers, see Classic Expressions.

   The following choices appear for the HTTP protocol:

   - **METHOD.** Filters HTTP requests that use a particular HTTP method.
   - **URL**. Filters HTTP requests for a specific Web page.
   - **URLQUERY.** Filters HTTP requests that contain a particular query string.
   - **VERSION.** Filters HTTP requests on the basis of the specified HTTP protocol version.
   - **HEADER.** Filters on the basis of a particular HTTP header.
   - **URLLEN.** Filters on the basis of the length of the URL.
   - **URLQUERY.** Filters on the basis of the query portion of the URL.
   - **URLQUERYLEN.** Filters on the basis of the length of the query portion of the URL only.

6. Under **Operator**, click the down arrow and choose the operator for your policy expression. For a complete list of choices see the “Operators” table in Classic Expressions. Some common operators are:

Operator	Description
==	Matches the specified value exactly or is exactly equal to the specified value.
!=	Does not match the specified value.
>	Is greater than the specified value.
<	Is less than the specified value.
>=	Is greater than or equal to the specified value.

<=	Is less than or equal to the specified value.
CONTAINS	Contains the specified value.
CONTENTS	Returns the contents of the designated header, URL, or URL query.
EXISTS	The specified header or query exists.
NOTCONTAINS	Does not contain the specified value.
NOTEXISTS	The specified header or query does not exist.

7. If a **Value** text box appears, type a string or numeric value, as appropriate. For example, chose **REQ** as the **Flow Type**, **HTTP** as the **Protocol**, and **HEADER** as the qualifier, and then type the value of the header string in the **Value** field and the header type for which you want to match the string in the **Header Name** text box.

8. Click **OK**.

9. To create a compound expression, click **Add**. Note that the type of compounding that is done depends on the following choices in the **Create Policy** dialog box:

   - **Match Any Expression.** The expressions are in a logical OR relationship.
   - **Match All Expressions.** The expressions are in a logical AND relationship.
   - **Tabular Expressions.** Click the **AND, OR,** and parentheses buttons to control evaluation.
   - **Advanced Free-Form.** Enter the expressions components directly into the Expression field, and click the **AND, OR,** and parentheses buttons to control evaluation.

Binding a Classic Policy

Depending on the policy type, you can bind a classic policy either globally or to a virtual server. Policy bind points are described in the table, Policy Type and Bind Points for Policies in Features That Use Classic Policies.

Note: You can bind a classic policy to multiple bind points.

To bind a classic policy globally by using the NetScaler command line

At the NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- bind <featureName> global <policyName> [**-priority** <positive_integer>]
- show <featureName> global

Example

```
> bind cmp global cmp-pol-compress -priority 2
 Done
> show cmp global
1)      Policy Name: cmp-pol-compress   Priority: 2
2)      Policy Name: ns_nocmp_xml_ie    Priority: 8700
3)      Policy Name: ns_nocmp_mozilla_47        Priority: 8800
4)      Policy Name: ns_cmp_mscss       Priority: 8900
5)      Policy Name: ns_cmp_msapp       Priority: 9000
6)      Policy Name: ns_cmp_content_type        Priority: 10000
 Done
>
```

To bind a classic policy to a virtual server by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- bind <featureName> vserver <name> [<targetVserver>] [**-policyName** <string> [**-priority** <positive_integer>]

- show <featureName> policy <name>

Example

```
> bind lb vserver lbtemp -policyName cmp-pol-compress -priority 1
 Done
> show lb vserver lbtemp
        lbtemp (10.102.29.101:80) - HTTP        Type: ADDRESS
        State: UP
        Last state change was at Tue Oct 27 06:40:38 2009 (+557 ms)
        Time since last state change: 0 days, 02:00:40.330
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state changed to UP
        Group: vserver-grp
        Mode: IP
        Persistence: COOKIEINSERT (version 0)   Persistence Backup: SOURCEIP   Persistence Mask: 255.255.255
        Persistence Timeout: 2 min      Backup Persistence Timeout: 2 min
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
1) http-one (10.102.29.252: 80) - HTTP State: UP       Weight: 1
        Persistence Cookie Value : NSC_wtfswfs-hsq=ffffffff096e03ed45525d5f4f58455e445a4a423660
1)      Policy : cmp-pol-compress Priority:1
 Done
>
```

Parameters for binding a classic policy

featureName

The name of the feature for which you are creating a policy. For Application Firewall policies, type `appfw`. For Access Gateway policies, type `accessgw`. For SSL policies, type `ssl`.

policyName

The name of the policy that you want to bind.

name

The name of the virtual server to which you bind the policy.

priority

The priority that you want to assign to the policy.

To bind a classic policy globally by using the configuration utility

Note: This procedure documents the Global Bindings dialog box. Depending on the feature for which you want to globally bind a policy, the route by which you arrive at this dialog box may be different.

1. In the navigation pane, expand the feature for which you want to globally bind a classic policy, and then locate the policy that you want to bind globally.

   Note: You cannot globally bind policies for Content Switching, Cache Redirection, SureConnect, Priority Queuing, or Access Gateway Authorization.

2. In the details pane, click **Global Bindings**.

3. In the **Bind/Unbind** <feature name> **Policy(s)** to Global dialog box, click **Insert Policy**.

4. In the **Policy Name** column, click the name of an existing policy that you want to globally bind, or click **New Policy** to open the **Create** <feature name> **Policy** dialog box.

5. After you have selected the policy or created a new policy, in the **Priority** column, type the priority value.

   The lower the number, the sooner this policy is applied relative to other policies. For example, a policy assigned a priority of 10 is applied before a policy with a priority of 100. You can use the same priority for different policies. All features that use classic policies implement only the first policy that a connection matches, so policy priority is important for getting the results you intend.

   As a best practice, leave room to add policies by setting priorities with intervals of 50 (or 100) between each policy.

6. Click **OK**.

To bind a classic policy to a virtual server by using the configuration utility

1. In the navigation pane, expand the feature that contains the virtual server to which you want to bind a classic policy (for example, if you want to bind a classic policy to a content switching virtual server, expand **Content Switching**), and then click **Virtual Servers**.

2. In the details pane, select the virtual server, and then click **Open**.

3. In the **Configure** <Feature> **Virtual Server** dialog box, on the **Policies** tab, click the feature icon for the type policy that you want, and then click **Insert Policy**.

4. In the **Policy Name** column, click the name of an existing policy that you want to bind to a virtual server, or click A to open the **Create** <feature name> **Policy** dialog box.

5. After you have selected the policy or created a new policy, in the **Priority** column, set the priority.

   If you are binding a policy to a content switching virtual server, in the **Target** column, select a load balancing virtual server to which traffic that matches the policy should be sent.

6. Click **OK**.

Viewing Classic Policies

You can view classic policies by using either the configuration utility or the command line. You can view details such as the policy's name, expression, and bindings.

To view a classic policy and its binding information by using the NetScaler command line

At the NetScaler command prompt, type the following commands to view a classic policy and its binding information:

show <featureName> policy [policyName]

Example

```
> show appfw policy GenericApplicationSSL_
      Name: GenericApplicationSSL_    Rule: ns_only_get_adv
      Profile: GenericApplicationSSL_Prof1    Hits: 0
      Undef Hits: 0
      Policy is bound to following entities
      1) REQ VSERVER app_u_GenericApplicationSSLPortalPages   PRIORITY : 100
 Done
```

Note: If you omit the policy name, all policies are listed without the binding details.

Parameters for viewing a classic policy

featureName

The name of the feature with which the policy is associated.

policyName

The name of the policy that you want to view.

To view classic policies and policy bindings by using the configuration utility

1. In the navigation pane, expand the feature whose policies you want to view, (for example, if you want to view Application Firewall policies, expand **Application Firewall**), and then click **Policies**.

2. In the details pane, do one or more of the following:

   - To view details for a specific policy, click the policy. Details appear in the **Details** area of the configuration pane.

   - To view bindings for a specific policy, click the policy, and then click **Show Bindings**.

   - To view global bindings, click the policy, and then click **Global Bindings**. Note that you cannot bind a Content Switching, Cache Redirection, SureConnect, Priority Queuing, or Access Gateway Authorization policy globally.

Creating Named Classic Expressions

A named classic expression is a classic expression that can be referenced through an assigned name. Often, you need to configure classic expressions that are large or complex and form a part of a larger compound expression. You might also configure classic expressions that you need to use frequently and in multiple compound expressions or classic policies. In these scenarios, you can create the classic expression you want, save it with a name of your choice, and then reference the expression from compound expressions or policies through its name. This saves configuration time and improves the readability of complex compound expressions. Additionally, any modifications to a named classic expression need to be made only once.

Some named expressions are built-in, and a subset of these are read-only. Built-in named expressions are divided into four categories: General, Anti-Virus, Personal Firewall, and Internet Security. General named expressions have a wide variety of uses. For example, from the General category, you can use the expressions `ns_true` and `ns_false` to specify a value of TRUE or FALSE, respectively, to be returned for all traffic. You can also identify data of a particular type (for example, HTM, DOC, or GIF files), determine whether caching headers are present, or determine whether the round trip time for packets between a client and the NetScaler is high (over 80 milliseconds).

Anti-Virus, Personal Firewall, and Internet Security named expressions test clients for the presence of a particular program and version and are used primarily in Access Gateway policies.

For descriptions of the built-in named expressions, see Classic Expressions"Classic Expressions".

Note: You cannot modify or delete built-in named expressions.

To create a named classic expression by using the NetScaler command line

At a NetScaler command prompt, type the following commands to set the parameters and verify the configuration:

- add expression <name> <value> [**-comment** <string>] [**-clientSecurityMessage** <string>]
- show expression [<name> | **-type** CLASSIC

Example

```
> add expression classic_ne "REQ.HTTP.URL CONTAINS www.example1.com" -comment "Checking the URL for
 Done
```

```
> show expression classic_ne
1)      Name: classic_ne  Expr: REQ.HTTP.URL CONTAINS www.example1.com  Hits: 0 Type : CLASSIC
        Comment: "Checking the URL for www.example1.com"
 Done
>
```

Parameters for creating a named classic expression

name

The name of the expression that will be created. This is a required argument. The maximum length of the expression is 63 characters.

value

The expression string. This is a required argument. The maximum length is 1499 characters.

comment

Any comments that you may want to associate with the expression. The maximum length is 255 characters.

clientSecurityMessage

The client security message that must be displayed if the expression evaluates to false. This parameter is valid for expressions that perform endpoint checks only. The maximum length is 127 characters.

To create a named classic expression by using the configuration utility

1. In the navigation pane, expand **AppExpert**, expand **Expressions**, and then click **Classic Expressions**.

2. In the details pane, click **Add**.

   Note: Some of the built-in expressions in the Expressions list are read-only.

3. In the **Create Policy Expression** dialog box, specify values for the following parameters, which correspond to parameters described in "Parameters for creating a named classic expression" as shown:

   - **Expression Name***—name
   - **Client Security Message**—clientSecurityMessage
   - **Comments**—comment

   * A required parameter

4. To create the expression, do one of the following:

   - You can choose inputs to this expression from the **Named Expressions** drop-down list.
   - You can create a new expression, as described in To add an expression for a classic policy by using the configuration utility.

5. When you are done, click **Close**. Verify that your new expression was created by scrolling to the bottom of the **Classic Expressions** list to view it.

Expressions Reference

The following tables list expressions and expression elements that you can use to identify specific types of data. The first table applies to default syntax expressions, in alphabetic order. The remaining tables cover the different types of classic expressions.

Default Syntax Expressions

The following table is a listing of default syntax expression prefixes, with cross-references to descriptions of these prefixes and the operators that you can specify for them. Note that some prefixes can work with multiple types of operators. For example, a cookie can be parsed by using operators for text or operators for HTTP headers.

You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

Note: The Description column in the following table contains cross-references to additional information about prefix usage and applicable operators for the prefix.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
`CLIENT.ETHER`	Prefixes for MAC Addresses. Operations for MAC Addresses.
`CLIENT.ETHER.[DSTMAC | SRCMAC]`	Prefixes for MAC Addresses. Operations for MAC Addresses.
`CLIENT.INTERFACE`	Designates an expression that refers to the ID of the network interface through which the current packet entered the Application Switch. See the other `CLIENT.INTERFACE` prefix descriptions in this table.
`CLIENT.INTERFACE.ID`	Extracts the ID of the network interface that received the current packet of data. See the other `CLIENT.INTERFACE` prefix descriptions in this table.
`CLIENT.INTERFACE.ID.EQ("id")`	Returns Boolean TRUE if the interface's ID matches the ID that is passed as the argument. For example: `CLIENT.INTERFACE.ID.EQ("1/1")` See Booleans in Compound Expressions.
`CLIENT.INTERFACE.[RXTHROU GHPUT | RXTXTHROUGHPUT | TXTHROUGHPUT]`	Expressions for Numeric Client and Server Data. Compound Operations for Numbers.
`CLIENT.IP`	Operates on the IP protocol data associated with the current packet. See the other `CLIENT.IP` prefixes in this table.

`CLIENT.IP.DST`	Prefixes for IPV4 Addresses and IP Subnets . Operations for IPV4 Addresses. Compound Operations for Numbers.
`CLIENT.IP.SRC`	Prefixes for IPV4 Addresses and IP Subnets . Operations for IPV4 Addresses. Compound Operations for Numbers.
`CLIENT.IPV6`	Operates on IPv6 protocol data. See the other `CLIENT.IPV6` prefixes in this table.
`CLIENT.IPV6.DST`	Expression Prefixes for IPv6 Addresses. Operations for IPV6 Prefixes.
`CLIENT.IPV6.SRC`	Expression Prefixes for IPv6 Addresses. Operations for IPV6 Prefixes.
`CLIENT.SSL`	Operates on the SSL protocol data for the current packet. See the other `CLIENT.SSL` prefixes in this table.
`CLIENT.SSL.CIPHER_BITS`	Prefixes for Numeric Data in SSL Certificates. Compound Operations for Numbers.
`CLIENT.SSL.CIPHER_EXPORTA BLE`	Prefixes for Text-Based SSL and Certificate Data. Booleans in Compound Expressions.
`CLIENT.SSL.CLIENT_CERT`	Expressions for SSL Certificates. Expressions for SSL Certificate Dates.
`CLIENT.SSL.IS_SSL`	Prefixes for Text-Based SSL and Certificate Data. Booleans in Compound Expressions.
`CLIENT.SSL.VERSION`	Prefixes for Numeric Data in SSL Certificates. Compound Operations for Numbers.
`CLIENT.TCP`	Operates on TCP protocol data. See the other `CLIENT.TCP` prefixes in this table.
`CLIENT.TCP.[DSTPORT | MSS | SRCPORT]`	Expressions for TCP, UDP, and VLAN Data. Compound Operations for Numbers.
`CLIENT.TCP.PAYLOAD(` *`integer`* `)`	Expressions for TCP, UDP, and VLAN Data. Default Syntax Expressions: Evaluating Text.

`CLIENT.UDP`	Operates on the UDP protocol data associated with the current packet. See the other `CLIENT.UDP` prefixes in this table.
`CLIENT.UDP.DNS.DOMAIN`	Expressions for TCP, UDP, and VLAN Data. Default Syntax Expressions: Evaluating Text.
`CLIENT.UDP.DNS.DOMAIN.EQ("`*`hostname`*`")`	Expressions for TCP, UDP, and VLAN Data. Booleans in Compound Expressions.
`CLIENT.UDP.DNS. [IS_AAAAREC | IS_ANYREC | IS_AREC | IS_CNAMEREC | IS_MXREC | IS_NSREC | IS_PTRREC | IS_SOAREC | IS_SRVREC]`	Expressions for TCP, UDP, and VLAN Data. Booleans in Compound Expressions.
`CLIENT.UDP.[DSTPORT | SRCPORT]`	Expressions for TCP, UDP, and VLAN Data. Compound Operations for Numbers.
`CLIENT.VLAN`	Operates on the VLAN through which the current packet entered the NetScaler. See the other `CLIENT.VLAN` prefixes in this table.
`CLIENT.VLAN.ID`	Expressions for TCP, UDP, and VLAN Data. Compound Operations for Numbers.
`HTTP.REQ`	Operates on HTTP requests. See the other `HTTP.REQ` prefixes in this table.
`HTTP.REQ.BODY(`*`integer`*`)`	Expression Prefixes for Text in HTTP Requests and Responses. Basic Operations on Text.
`HTTP.REQ.CACHE_CONTROL`	Prefixes for Cache-Control Headers. Operations for Cache-Control Headers.
`HTTP.REQ.CONTENT_LENGTH`	Expressions for Numeric HTTP Payload Data Other Than Dates. Compound Operations for Numbers.
`HTTP.REQ.COOKIE`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.

`HTTP.REQ.DATE`	Format of Dates and Times in an Expression. Expressions for HTTP Request and Response Dates. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers. Operations for HTTP Headers.
`HTTP.REQ.HEADER("header_name")`	Expression Prefixes for Text in HTTP Requests and Responses. Prefixes for HTTP Headers. Operations for HTTP Headers.
`HTTP.REQ.FULL_HEADER("header_name")`	Prefixes for HTTP Headers. Operations for HTTP Headers.
`HTTP.REQ.HOSTNAME`	Expression Prefixes for Text in HTTP Requests and Responses.
`HTTP.REQ.HOSTNAME.[DOMAIN | Server]`	Expression Prefixes for Text in HTTP Requests and Responses. Basic Operations on Text.
`HTTP.REQ.HOSTNAME.EQ("hostname")`	Expression Prefixes for Text in HTTP Requests and Responses. Booleans in Compound Expressions. “Basic Operations on Expression Prefixes”.
`HTTP.REQ.HOSTNAME.PORT`	Expression Prefixes for Text in HTTP Requests and Responses. Compound Operations for Numbers.
`HTTP.REQ.IS_VALID`	Returns TRUE if the HTTP request is properly formed. See Booleans in Compound Expressions.
`HTTP.REQ.METHOD`	Expression Prefixes for Text in HTTP Requests and Responses. Basic Operations on Text. Complex Operations on Text.
`HTTP.REQ.TRACKING`	Returns the HTTP body tracking mechanism. See the descriptions of other `HTTP.REQ.TRACKING` prefixes in this table.
`HTTP.REQ.TRACKING.EQ("tracking_mechanism")`	Returns TRUE or FALSE. See Booleans in Compound Expressions.

`HTTP.REQ.URL`	Obtains the HTTP URL object from the request and sets the text mode to URLENCODED by default. See Expression Prefixes for Text in HTTP Requests and Responses.
`HTTP.REQ.URL.[CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX | VERSION]`	Expression Prefixes for Text in HTTP Requests and Responses. Basic Operations on Text. Complex Operations on Text.
`HTTP.REQ.URL.HOSTNAME.EQ("hostname")`	Expression Prefixes for Text in HTTP Requests and Responses. Booleans in Compound Expressions.
`HTTP.REQ.URL.HOSTNAME.PORT`	Expression Prefixes for Text in HTTP Requests and Responses. Compound Operations for Numbers.
`HTTP.REQ.URL.PATH.IGNORE_EMPTY_ELEMENTS`	Ignores spaces in the data. See the table HTTP Expression Prefixes that Return Text.
`HTTP.REQ.URL.QUERY.IGNORE_EMPTY_ELEMENTS`	Ignores spaces in the data. See the table HTTP Expression Prefixes that Return Text.
`HTTP.REQ.USER.IS_MEMBER_OF`	HTTP Expression Prefixes that Return Text.
`HTTP.REQ.USER.NAME`	HTTP Expression Prefixes that Return Text.
`HTTP.REQ.VERSION`	Expression Prefixes for Text in HTTP Requests and Responses.
`HTTP.REQ.VERSION.[MAJOR | MINOR]`	Operates on the major or minor HTTP version string. See Expression Prefixes for Text in HTTP Requests and Responses and Compound Operations for Numbers.
`HTTP.RES`	Operates on HTTP responses.
`HTTP.RES.BODY(integer)`	Expression Prefixes for Text in HTTP Requests and Responses Basic Operations on Text. Complex Operations on Text.
`HTTP.RES.CACHE_CONTROL`	Prefixes for Cache-Control Headers. Operations for Cache-Control Headers.
`HTTP.RES.CONTENT_LENGTH`	Expression Prefixes for Text in HTTP Requests and Responses. Operations for HTTP Headers. Compound Operations for Numbers.

`HTTP.RES.DATE`	Format of Dates and Times in an Expression. Expressions for HTTP Request and Response Dates. Expression Prefixes for Text in HTTP Requests and Responses. Compound Operations for Numbers. Operations for HTTP Headers.
`HTTP.RES.HEADER("`*`header_name`*`")`	Expression Prefixes for Text in HTTP Requests and Responses. Prefixes for HTTP Headers. Operations for HTTP Headers.
`HTTP.REQ.FULL_HEADER("`*`header_name`*`")`	Prefixes for HTTP Headers. Operations for HTTP Headers.
`HTTP.REQ.TXID`	Prefixes for HTTP Headers. Operations for HTTP Headers.
`HTTP.RES.IS_VALID`	Returns TRUE if the HTTP response is properly formed. See Booleans in Compound Expressions.
`HTTP.RES.SET_COOKIE`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE.COOKIE("`*`name`*`")`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE.COOKIE.[DOMAIN | PATH | PORT ]`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.

`HTTP.RES.SET_COOKIE.COOKIE.EXPIRES`	Obtains the Expires field of the cookie as a date string. The value of the Expires attribute can be operated upon as a time object. If multiple Expires fields are present, this expression operates on the first one. If the Expires attribute is absent, a string of length zero is returned. Also see: Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE.COOKIE.PATH.IGNORE_SPACES_IN_TEXT`	Ignores spaces in the data. For an example, see the table Expression Prefixes for Text in HTTP Requests and Responses.
`HTTP.RES.SET_COOKIE.COOKIE.PORT.IGNORE_SPACES_IN_TEXT`	Ignores spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text.
`HTTP.RES.SET_COOKIE.COOKIE.VERSION`	Prefixes for HTTP Headers. Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE.COOKIE("name"[, integer])[.PORT | PATH | DOMAIN | VERSION | EXPIRES]`	Prefixes for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE.COOKIE.EXPIRES`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE.EXISTS("name")`	Prefixes for HTTP Headers. Booleans in Compound Expressions.
`HTTP.RES.SET_COOKIE2`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE2.COOKIE("name")`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.

`HTTP.RES.SET_COOKIE2.COOKIE.[DOMAIN | PATH | PORT ]`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE2.COOKIE.EXPIRES`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE2.COOKIE.PATH.IGNORE_EMPTY_ELEMENTS`	Ignore spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text.
`HTTP.RES.SET_COOKIE2.COOKIE.PORT.IGNORE_EMPTY_ELEMENTS`	Ignore spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text. See also Default Syntax Expressions: Evaluating Text and Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE2.COOKIE("name"[,integer])[.HTTP | PATH | DOMAIN | VERSION | EXPIRES]`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE2.COOKIE.DOMAIN`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text.
`HTTP.RES.SET_COOKIE2.COOKIE.EXPIRES`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE2.COOKIE.VERSION`	Prefixes for HTTP Headers. Operations for HTTP Headers. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers.
`HTTP.RES.SET_COOKIE2.EXISTS("name")`	Prefixes for HTTP Headers. Operations for HTTP Headers. Booleans in Compound Expressions.

`HTTP.RES.STATUS`	Expression Prefixes for Text in HTTP Requests and Responses. Compound Operations for Numbers.
`HTTP.RES.STATUS_MSG`	Expression Prefixes for Text in HTTP Requests and Responses.
`HTTP.RES.TRACKING`	Returns the HTTP body tracking mechanism. See the descriptions of other `HTTP.REQ.TRACKING` prefixes in this table.
`HTTP.RES.TRACKING.EQ("tracking_method")`	Returns TRUE or FALSE. See Booleans in Compound Expressions.
`HTTP.RES.TXID`	Prefixes for HTTP Headers. Operations for HTTP Headers.
`HTTP.RES.VERSION`	Expression Prefixes for Text in HTTP Requests and Responses.
`HTTP.RES.VERSION.[MAJOR | MINOR]`	Operates on the major or minor HTTP version string. See Expression Prefixes for Text in HTTP Requests and Responses and Compound Operations for Numbers.
`SERVER`	Designates an expression that refers to the server. This is the starting point for access into parameters such as Ether and SSL. See the other SERVER prefixes in this table.
`SERVER.ETHER`	Operates on the ethernet protocol data associated with the current packet. See the other SERVER prefixes in this table.
`SERVER.ETHER.DSTMAC`	Prefixes for MAC Addresses. Prefixes for MAC Addresses.
`SERVER.INTERFACE`	Designates an expression that refers to the ID of the network interface that received the current packet of data. See the other `SERVER.INTERFACE` prefixes in this table.
`SERVER.INTERFACE.ID.EQ("id")`	Returns Boolean TRUE if the interface's ID matches the ID that is passed as the argument. For example: `SERVER.INTERFACE.ID.EQ("LA/1")` See Booleans in Compound Expressions.
`SERVER.INTERFACE.[RXTHROUGHPUT | RXTXTHROUGHPUT | TXTHROUGHPUT]`	Expressions for Numeric Client and Server Data. Compound Operations for Numbers.
`SERVER.IP`	Operates on the IP protocol data associated with the current packet. See the other `SERVER.IP` prefixes in this table.

SERVER.IP.[DST | SRC]	Prefixes for IPV4 Addresses and IP Subnets. Operations for IPV4 Addresses. Compound Operations for Numbers.
SERVER.IPV6	Operates on IPv6 protocol data. See the other SERVER.IPV6 prefixes in this table.
SERVER.IPV6.DST	Expression Prefixes for IPv6 Addresses. Operations for IPV6 Prefixes.
SERVER.IPV6.SRC	Expression Prefixes for IPv6 Addresses. Operations for IPV6 Prefixes.
SERVER.TCP	Operates on TCP protocol data. See the other CLIENT.TCP prefixes in this table.
SERVER.TCP.[DSTPORT | MSS | SRCPORT]	Expressions for TCP, UDP, and VLAN Data. Compound Operations for Numbers.
SERVER.VLAN	Operates on the VLAN through which the current packet entered the NetScaler. See the other SERVER.VLAN prefixes in this table.
SERVER.VLAN.ID	Expressions for TCP, UDP, and VLAN Data. Compound Operations for Numbers.
SYS	Designates an expression that refers to the NetScaler itself, not to the client or server.. See the other SYS prefixes in this table.
SYS.EVAL_CLASSIC_EXPR(*classic_expression*)	Classic Expressions in Default Syntax Expressions. Booleans in Compound Expressions.
SYS.HTTP_CALLOUT(*http_callout*)	HTTP Callouts.
SYS.CHECK_LIMIT	Rate Limiting.
SYS.TIME	Expressions for the NetScaler System Time. Compound Operations for Numbers.
SYS.TIME.[BETWEEN(*time1, time2*) | EQ(*time*) | GE(*time*) | GT(*time*) | LE(*time*) | LT(*time*) | WITHIN(*time1, time2*)]	Expressions for the NetScaler System Time. Booleans in Compound Expressions. Compound Operations for Numbers.
SYS.TIME.[DAY | HOURS | MINUTES | MONTH | RELATIVE_BOOT | RELATIVE_NOW SECONDS | WEEKDAY | YEAR]	Expressions for the NetScaler System Time. Compound Operations for Numbers.

`VPN.BASEURL.[CVPN_DECODE | CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | HOSTNAME.SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX]`	Expression Prefixes for VPNs and Clientless VPNs.
`VPN.BASEURL.HOSTNAME.EQ("hostname")`	Expression Prefixes for VPNs and Clientless VPNs. Booleans in Compound Expressions.
`VPN.BASEURL.HOSTNAME.PORT`	Expression Prefixes for VPNs and Clientless VPNs. Compound Operations for Numbers.
`VPN.BASEURL.PATH.IGNORE_EMPTY_ELEMENTS`	Ignores spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text.
`VPN.BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS`	Ignores spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text.
`VPN.CLIENTLESS_BASEURL`	Expression Prefixes for VPNs and Clientless VPNs.
`VPN.CLIENTLESS_BASEURL.[CVPN_DECODE | CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | HOSTNAME.SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX]`	Expression Prefixes for VPNs and Clientless VPNs.
`VPN.CLIENTLESS_BASEURL.HOSTNAME.EQ("hostname")`	Expression Prefixes for VPNs and Clientless VPNs. Booleans in Compound Expressions.
`VPN.CLIENTLESS_BASEURL.HOSTNAME.PORT`	Expression Prefixes for VPNs and Clientless VPNs. Compound Operations for Numbers.
`VPN.CLIENTLESS_BASEURL.PATH.IGNORE_EMPTY_ELEMENTS`	Ignores spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text.
`VPN.CLIENTLESS_BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS`	Ignores spaces in the data. For an example, see the table HTTP Expression Prefixes that Return Text.
`VPN.CLIENTLESS_HOSTURL`	Expression Prefixes for VPNs and Clientless VPNs.

`VPN.CLIENTLESS_HOSTURL.[CVPN_DECODE | CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | HOSTNAME.SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX]`	Expression Prefixes for VPNs and Clientless VPNs.
`VPN.CLIENTLESS_HOSTURL.HOSTNAME.EQ("hostname")`	Expression Prefixes for VPNs and Clientless VPNs. Booleans in Compound Expressions.
`VPN.CLIENTLESS_HOSTURL.HOSTNAME.PORT`	Expression Prefixes for VPNs and Clientless VPNs. Compound Operations for Numbers.
`VPN.CLIENTLESS_HOSTURL.PATH.IGNORE_EMPTY_ELEMENTS`	Ignores empty elements in data. For an example, see the table HTTP Expression Prefixes that Return Text.
`VPN.CLIENTLESS_HOSTURL.QUERY.IGNORE_EMPTY_ELEMENTS`	Ignores empty elements in data. For an example, see the table HTTP Expression Prefixes that Return Text.
`VPN.HOST`	Expression Prefixes for VPNs and Clientless VPNs.
`VPN.HOST.[DOMAIN | Server]`	Expression Prefixes for VPNs and Clientless VPNs.
`VPN.HOST.EQ("hostname")`	Expression Prefixes for VPNs and Clientless VPNs. Booleans in Compound Expressions.
`VPN.HOST.PORT`	Expression Prefixes for VPNs and Clientless VPNs. Default Syntax Expressions: Evaluating Text. Compound Operations for Numbers.

Classic Expressions

The subtopics listed in the table of contents on the left side of your screen contain tables listing the NetScaler classic expressions.

In the table of operators, the result type of each operator is shown at the beginning of the description. In the other tables, the level of each expression is shown at the beginning of the description. For named expressions, each expression is shown as a whole.

Operators

Expression Element	Definition
==	Boolean. Returns TRUE if the current expression equals the argument. For text operations, the items being compared must exactly match one another. For numeric operations, the items must evaluate to the same number.
!=	Boolean. Returns TRUE if the current expression does not equal the argument. For text operations, the items being compared must not exactly match one another. For numeric operations, the items must not evaluate to the same number.
CONTAINS	Boolean. Returns TRUE if the current expression contains the string that is designated in the argument.
NOTCONTAINS	Boolean. Returns TRUE if the current expression does not contain the string that is designated in the argument.
CONTENTS	Text. Returns the contents of the current expression.
EXISTS	Boolean. Returns TRUE if the item designated by the current expression exists.
NOTEXISTS	Boolean. Returns TRUE if the item designated by the current expression does not exist.

>	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than the argument.
<	Boolean. Returns TRUE if the current expression evaluates to a number that is less than the argument.
>=	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than or equal to the argument.
<=	Boolean. Returns TRUE if the current expression evaluates to a number that is less than or equal to the argument.

General Expressions

Expression Element	Definition
`REQ`	Flow Type. Operates on incoming (or request) packets.
`REQ.HTTP`	Protocol Operates on HTTP requests.
`REQ.HTTP.METHOD`	Qualifier Designates the HTTP method.
`REQ.HTTP.URL`	Qualifier Designates the URL.
`REQ.HTTP.URLTOKENS`	Qualifier Designates the URL token.
`REQ.HTTP.VERSION`	Qualifier Designates the HTTP version.
`REQ.HTTP.HEADER`	Qualifier Designates the HTTP header.
`REQ.HTTP.URLLEN`	Qualifier Designates the number of characters in the URL.
`REQ.HTTP.URLQUERY`	Qualifier Designates the query portion of the URL.
`REQ.HTTP.URLQUERYLEN`	Qualifier Designates the length of the query portion of the URL.
`REQ.SSL`	Protocol Operates on SSL requests.
`REQ.SSL.CLIENT.CERT`	Qualifier Designates the entire client certificate.

REQ.SSL.CLIENT.CERT.SUBJECT	Qualifier Designates the client certificate subject.
REQ.SSL.CLIENT.CERT.ISSUER	Qualifier Designates the issuer of the client certificate.
REQ.SSL.CLIENT.CERT.SIGALGO	Qualifier Designates the validation algorithm used by the client certificate.
REQ.SSL.CLIENT.CERT.VERSION	Qualifier Designates the client certificate version.
REQ.SSL.CLIENT.CERT.VALIDFROM	Qualifier Designates the date before which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.VALIDTO	Qualifier Designates the date after which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.SERIALNUMBER	Qualifier Designates the serial number of the client certificate.
REQ.SSL.CLIENT.CIPHER.TYPE	Qualifier Designates the encryption protocol used by the client.
REQ.SSL.CLIENT.CIPHER.BITS	Qualifier Designates the number of bits used by the client’s SSL key.
REQ.SSL.CLIENT.SSL.VERSION	Qualifier Designates the SSL version that the client is using.
REQ.TCP	Protocol Operates on incoming TCP packets.
REQ.TCP.SOURCEPORT	Qualifier Designates the source port of the incoming packet.
REQ.TCP.DESTPORT	Qualifier Designates the destination port of the incoming packet.

REQ.IP	Protocol Operates on incoming IP packets.
REQ.IP.SOURCEIP	Qualifier Designates the source IP of the incoming packet.
REQ.IP.DESTIP	Qualifier Designates the destination IP of the incoming packet.
RES	Flow Type Operates on outgoing (or response) packets.
RES.HTTP	Protocol Operates on HTTP responses.
RES.HTTP.VERSION	Qualifier Designates the HTTP version.
RES.HTTP.HEADER	Qualifier Designates the HTTP header.
RES.HTTP.STATUSCODE	Qualifier Designates the status code of the HTTP response.
RES.TCP	Protocol Operates on incoming TCP packets.
RES.TCP.SOURCEPORT	Qualifier Designates the source port of the outgoing packet.
RES.TCP.DESTPORT	Qualifier Designates the destination port of the outgoing packet.
RES.IP	Protocol Operates on outgoing IP packets.

RES.IP.SOURCEIP	Qualifier Designates the source IP of the outgoing packet. This can be in IPv4 or IPv6 format. For example: `add expr exp3 “sourceip == 10.102.32.123 –netmask 255.255.255.0 && destip == 2001::23/120”.`
RES.IP.DESTIP	Qualifier Designates the destination IP of the outgoing packet.

Client Security Expressions

Actual Expression	Definition
`CLIENT.APPLICATION.AV(<NAME>.VERSION == <VERSION>)`	Checks whether the client is running the designated anti-virus program and version.
`CLIENT.APPLICATION.AV(<NAME>.VERSION != <VERSION>)`	Checks whether the client is not running the designated anti-virus program and version.
`CLIENT.APPLICATION.PF(<NAME>.VERSION == <VERSION>)`	Checks whether the client is running the designated personal firewall program and version.
`CLIENT.APPLICATION.PF(<NAME>.VERSION != <VERSION>)`	Checks whether the client is not running the designated personal firewall program and version.
`CLIENT.APPLICATION.IS(<NAME>.VERSION == <VERSION>)`	Checks whether the client is running the designated internet security program and version.
`CLIENT.APPLICATION.IS(<NAME>.VERSION != <VERSION>)`	Checks whether the client is not running the designated internet security program and version.
`CLIENT.APPLICATION.AS(<NAME>.VERSION == <VERSION>)`	Checks whether the client is running the designated anti-spam program and version.
`CLIENT.APPLICATION.AS(<NAME>.VERSION != <VERSION>)`	Checks whether the client is not running the designated anti-spam program and version.

Network-Based Expressions

Expression	Definition
REQ	Flow Type. Operates on incoming, or request, packets.
REQ.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
REQ.INTERFACE.ID	Qualifier. Operates on the ID of the designated NetScaler interface.
REQ.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated NetScaler interface.
REQ.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated NetScaler interface.
REQ.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated NetScaler interface.
REQ.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
REQ.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.
RES	Flow Type. Operates on outgoing (or response) packets.
RES.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.

RES.INTERFACE.ID	Qualifier. Operates on the ID of the designated NetScaler interface.
RES.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated NetScaler interface.
RES.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated NetScaler interface.
RES.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated NetScaler interface.
RES.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
RES.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.

Date/Time Expressions

Expression	Definition
TIME	Qualifier. Operates on the date and time of day, GMT.
DATE	Qualifier. Operates on the date, GMT.
DAYOFWEEK	Operates on the specified day in the week, GMT.

File System Expressions

You can specify file system expressions in authorization policies for users and groups who access file sharing through the Access Gateway file transfer utility (the VPN portal). These expressions work with the Access Gateway's file transfer authorization feature to control user access to file servers, folders, and files. For example, you can use these expressions in authorization policies to control access based on file type and size.

Expression	Definition
`FS.COMMAND`	Qualifier. Operates on a file system command. The user can issue multiple commands on a file transfer portal. (For example, ls to list files or mkdir to create a directory). This expression returns the current action that the user is taking. Possible values: Neighbor, login, ls, get, put, rename, mkdir, rmdir, del, logout, any. Following is an example: `Add authorization policy pol1 "fs.command eq login && (fs.user eq administrator || fs.serverip eq 10.102.88.221 –netmask 255.255.255.252)" allow`
`FS.USER`	Returns the user who is logged on to the file system.
`FS.SERVER`	Returns the host name of the target server. In the following example, the string `win2k3-88-22` is the server name: `fs.server eq win2k3-88-221`
`FS.SERVERIP`	Returns the IP address of the target server.

FS.SERVICE	Returns a shared root directory on the file server. If a particular folder is exposed as shared, a user can directly log on to the specified first level folder. This first level folder is called a service. For example, in the path \\hostname\SERVICEX\ETC, SERVICEX is the service. As another example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.SERVICE will return service1. Following is an example: `fs.service notcontains New`
FS.DOMAIN	Returns the domain name of the target server.
FS.PATH	Returns the complete path of the file being accessed. For example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.PATH will return \service\dir1\file1.doc. Following is an example: `fs.path notcontains SSL`
FS.FILE	Returns the name of the file being accessed. For example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.FILE will return file1.doc.
FS.DIR	Returns the directory being accessed. For example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.DIR will return \service\dir1.
FS.FILE.ACCESSTIME	Returns the time at which the file was last accessed. This is one of several options that provide you with granular control over actions that the user performs. (See the following entries in this table.)
FS.FILE.CREATETIME	Returns the time at which the file was created.
FS.FILE.MODIFYTIME	Returns the time at which the file was edited.
FS.FILE.WRITETIME	Returns the time of the most recent change in the status of the file.
FS.FILE.SIZE	Returns the file size.
FS.DIR.ACCESSTIME	Returns the time at which the directory was last accessed.
FS.DIR.CREATETIME	Returns the time at which the directory was created.

FS.DIR.MODIFYTIME	Returns the time at which the directory was last modified.
FS.DIR.WRITETIME	Returns the time at which the directory status last changed.

Note: File system expressions do not support regular expressions.

Built-In Named Expressions (General)

Expression	Definition
ns_all_apps_ncomp	Tests for connections with destination ports between 0 and 65535. In other words, tests for all applications.
ns_cachecontrol_nocache	Tests for connections with an HTTP Cache-Control header that contains the value “no-cache”.
ns_cachecontrol_nostore	Tests for connections with an HTTP Cache-Control header that contains the value “no-store”.
ns_cmpclient	Tests the client to determine if it accepts compressed content.
ns_content_type	Tests for connections with an HTTP Content-Type header that contains “text”.
ns_css	Tests for connections with an HTTP Content-Type header that contains “text/css”.
ns_ext_asp	Tests for HTTP connections to any URL that contains the string .asp—in other words, any connection to an active server page (ASP).
ns_ext_cfm	Tests for HTTP connections to any URL that contains the string .cfm
ns_ext_cgi	Tests for HTTP connections to any URL that contains the string .cgi—in other words, any connection to a common gateway interface (CGI) script.
ns_ext_ex	Tests for HTTP connections to any URL that contains the string .ex
ns_ext_exe	Tests for HTTP connections to any URL that contains the string .exe—in other words, any connection to a executable file.
ns_ext_htx	Tests for HTTP connections to any URL that contains the string .htx
ns_ext_not_gif	Tests for HTTP connections to any URL that does not contain the string .gif—in other words, any connection to a URL that is not a GIF image.

ns_ext_not_jpeg	Tests for HTTP connections to any URL that does not contain the string .jpeg—in other words, any connection to a URL that is not a JPEG image.
ns_ext_shtml	Tests for HTTP connections to any URL that contains the string .shtml—in other words, any connection to a server-parsed HTML page.
ns_false	Always returns a value of FALSE.
ns_farclient	Client is in a different geographical region from the NetScaler, as determined by the geographical region in the client's IP address. The following regions are predefined: 192.0.0.0 – 193.255.255.255: Multi-regional 194.0.0.0 – 195.255.255.255: European Union 196.0.0.0 – 197.255.255.255: Other1 198.0.0.0 – 199.255.255.255: North America 200.0.0.0 – 201.255.255.255: Central and South America 202.0.0.0 – 203.255.255.255: Pacific Rim 204.0.0.0 – 205.255.255.255: Other2 206.0.0.0 – 207.255.255.255: Other3
ns_header_cookie	Tests for HTTP connections that contain a Cookie header
ns_header_pragma	Tests for HTTP connections that contain a Pragma: no-cache header.
ns_mozilla_47	Tests for HTTP connections whose User-Agent header contains the string Mozilla/4.7—in other words, any connection from a client using the Mozilla 4.7 Web browser.
ns_msexcel	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msexcel—in other words, any connection transmitting a Microsoft Excel spreadsheet.

ns_msie	Tests for HTTP connections whose User-Agent header contains the string MSIE—in other words, any connection from a client using any version of the Internet Explorer Web browser.
ns_msppt	Tests for HTTP connections whose Content-Type header contains the string application/vnd.ms-powerpoint—in other words, any connection transmitting a Microsoft PowerPoint file.
ns_msword	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msword—in other words, any connection transmitting a Microsoft Word file.
ns_non_get	Tests for HTTP connections that use any HTTP method except for GET.
ns_slowclient	Returns TRUE if the average round trip time between the client and the NetScaler is more than 80 milliseconds.
ns_true	Returns TRUE for all traffic.
ns_url_path_bin	Tests the URL path to see if it points to the /bin/ directory.
ns_url_path_cgibin	Tests the URL path to see if it points to the CGI-BIN directory.
ns_url_path_exec	Tests the URL path to see if it points to the /exec/ directory.
ns_url_tokens	Tests for the presence of URL tokens.
ns_xmldata	Tests for the presence of XML data.

Built-In Named Expressions (Anti-Virus)

Expression	Definition
McAfee Virus Scan 11	Tests to determine whether the client is running the latest version of McAfee VirusScan.
Mcafee Antivirus	Tests to determine whether the client is running any version of McAfee Antivirus.
Symantec AntiVirus 10 (with Updated Definition File)	Tests to determine whether the client is running the most current version of Symantec AntiVirus.
Symantec AntiVirus 6.0	Tests to determine whether the client is running Symantec AntiVirus 6.0.
Symantec AntiVirus 7.5	Tests to determine whether the client is running Symantec AntiVirus 7.5.
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
TrendMicro AntiVirus 11.25	Tests to determine whether the client is running Trend Microsystems' AntiVirus, version 11.25.
Sophos Antivirus 4	Tests to determine whether the client is running Sophos Antivirus, version 4.
Sophos Antivirus 5	Tests to determine whether the client is running Sophos Antivirus, version 5.
Sophos Antivirus 6	Tests to determine whether the client is running Sophos Antivirus, version 6.

Built-In Named Expressions (Personal Firewall)

Expression	Definition
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
Sygate Personal Firewall 5.6	Tests to determine whether the client is running the Sygate Personal Firewall, version 5.6.
ZoneAlarm Personal Firewall 6.5	Tests to determine whether the client is running the ZoneAlarm Personal Firewall, version 6.5.

Built-In Named Expressions (Client Security)

Expression	Definition
Norton Internet Security	Tests to determine whether the client is running any version of Norton Internet Security.

Summary Examples of Default Syntax Expressions and Policies

The following table provides examples of default syntax expressions that you can use as the basis for your own default syntax expressions.

Table 1. Examples of Default Syntax Expressions

Expression Type	Sample Expressions

Look at the method used in the HTTP request.	`http.req.method.eq(post)`
	
	`http.req.method.eq(get)`

Check the Cache-Control or Pragma header value in an HTTP request (`req`) or response (`res`).	`http.req.header("Cache-Control").contains("no-store")`
	
	`http.req.header("Cache-Control").contains("no-cache")`
	
	`http.req.header("Pragma").contains("no-cache")`
	
	`http.res.header("Cache-Control").contains("private")`
	
	`http.res.header("Cache-Control").contains("public")`
	
	`http.res.header("Cache-Control").contains("must-revalidate")`
	
	`http.res.header("Cache-Control").contains("proxy-revalidate")`
	
	`http.res.header("Cache-Control").contains("max-age")`

Check for the presence of a header in a request (`req`) or response (`res`).	`http.req.header("myHeader").exists`
	
	`http.res.header("myHeader").exists`

Look for a particular file type in an HTTP request based on the file extension.	`http.req.url.contains(".html")`
	
	`http.req.url.contains(".cgi")`
	
	`http.req.url.contains(".asp")`
	
	`http.req.url.contains(".exe")`
	
	`http.req.url.contains(".cfm")`
	
	`http.req.url.contains(".ex")`
	
	`http.req.url.contains(".shtml")`
	
	`http.req.url.contains(".htx")`
	
	`http.req.url.contains("/cgi-bin/")`
	
	`http.req.url.contains("/exec/")`
	
	`http.req.url.contains("/bin/")`

Look for anything that is other than a particular file type in an HTTP request.	`http.req.url.contains(".gif").not`
	
	`http.req.url.contains(".jpeg").not`

Check the type of file that is being sent in an HTTP response based on the Content-Type header.	`http.res.header("Content-Type").contains("text")`
	
	`http.res.header("Content-Type").contains("application/msword")`
	
	`http.res.header("Content-Type").contains("vnd.ms-excel")`
	
	`http.res.header("Content-Type").contains("application/vnd.ms-powerpoint")`
	
	`http.res.header("Content-Type").contains("text/css")`
	
	`http.res.header("Content-Type").contains("text/xml")`
	
	`http.res.header("Content-Type").contains("image/")`

Check whether this response contains an expiration header.	`http.res.header("Expires").exists`
Check for a Set-Cookie header in a response.	`http.res.header("Set-Cookie").exists`

Check the agent that sent the response.	`http.res.header("User-Agent").contains("Mozilla/4.7")`
	
	`http.res.header("User-Agent").contains("MSIE")`

Check if the first 1024 bytes of the body of a request starts with the string “some text”.	`http.req.body(1024).contains("some text")`

The following table shows examples of policy configurations and bindings for commonly used functions.

Table 2. Examples of Default Syntax Expressions and Policies

Purpose	Example

Use the rewrite feature to replace occurrences of `http:// with https://` in the body of an HTTP response.	`add rewrite action httpRewriteAction replace_all http.res.body(50000) "\"https://\"" -pattern http://`
	
	`add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction`

Replace all occurrences of “abcd” with “1234” in the first 1000 bytes of the HTTP body.	`add rewrite action abcdTo1234Action replace_all "http.req.body(1000)" "\"1234\"" -pattern abcd`
	
	`add rewrite policy abcdTo1234Policy "http.req.body(1000).contains(\"abcd\")" abcdTo1234Action`
	
	`bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE`

Downgrade the HTTP version to 1.0 to prevent the server from chunking HTTP responses.	`add rewrite action downgradeTo1.0Action replace http.req.version.minor "\"0\""`
	
	`add rewrite policy downgradeTo1.0Policy "http.req.version.minor.eq(1)" downgradeTo1.0Action`
	
	`bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST`

Remove references to the HTTP or HTTPS protocol in all responses, so that if the user's connection is HTTP, the link is opened by using HTTP, and if the user's connection is HTTPS, the link is opened by using HTTPS.	`add rewrite action remove_http_https replace_all "http.res.body(1000000).set_text_mode(ignorecase)" "\"//\"" -pattern "re~https?://|HTTPS?://~"`
	
	`add rewrite policy remove_http_https true remove_http_https`
	
	`bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE`

Rewrite instances of `http:// to https://` in all URLs.
 	
This policy uses the responder functionality.
	`add responder action httpToHttpsAction redirect "\"https://\" + http.req.hostname + http.req.url" -bypassSafetyCheck YES`
	
	`add responder policy httpToHttpsPolicy "!CLIENT.SSL.IS_SSL" httpToHttpsAction`
	
	`bind responder global httpToHttpsPolicy 1 END -type OVERRIDE`

Modify a URL to redirect from URL A to URL B. In this example, “file5.html” is appended to the path.
 	
This policy uses the responder functionality.
	`add responder action appendFile5Action redirect "\"http://\" + http.req.hostname + http.req.url + \"/file5.html\"" -bypassSafetyCheck YES`
	
	`add responder policy appendFile5Policy "http.req.url.eq(\"/testsite\")" appendFile5Action`
	
	`bind responder global appendFile5Policy 1 END -type OVERRIDE`

Redirect an external URL to an internal URL.	`add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server' '"www.my.host.com"'`
	
	`add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")' act_external_to_internal`
	
	`bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE`

Redirect requests to www.example.com that have a query string to www.Web*n*.example.com. The value *n* is derived from a server parameter in the query string, for example, `server=5`.	`add rewrite action act_redirect_query REPLACE q#http.req.header("Host").before_str(".example.com")' '"Web" + http.req.url.query.value("server")#`
	
	`add rewrite policy pol_redirect_query q#http.req.header("Host").eq("www.example.com") && http.req.url.contains("?")' act_redirect_query#`

Limit the number of requests per second from a URL.	`add ns limitSelector ip_limit_selector http.req.url "client.ip.src"`
	
	`add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector`
	
	`add responder action my_Web_site_redirect_action redirect "\"http://www.mycompany.com/\""`
	
	`add responder policy ip_limit_responder_policy "http.req.url.contains(\"myasp.asp\") && sys.check_limit(\"ip_limit_identifier\")" my_Web_site_redirect_action`
	
	`bind responder global ip_limit_responder_policy 100 END -type default`

Check the client IP address but pass the request without modifying the request.	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' NOREWRITE`
	
	`bind rewrite global check_client_ip_policy 100 END`

Remove old headers from a request and insert an NS-Client header.	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for`
	
	`add rewrite action del_client_ip delete_http_header client-ip`
	
	`add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' del_x_forwarded_for`
	
	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip`
	
	`add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'`
	
	`add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' insert_ns_client_header`
	
	`bind rewrite global check_x_forwarded_for_policy 100 200`
	
	`bind rewrite global check_client_ip_policy 200 300`
	
	`bind rewrite global insert_ns_client_policy 300 END`

Remove old headers from a request, insert an NS-Client header, and then modify the “insert header” action so that the value of the inserted header contains the client IP values from the old headers and the NetScaler’s connection IP address.
 	
Note that this example repeats the previous example, with the exception of the final set `rewrite action`.
	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for`
	
	`add rewrite action del_client_ip delete_http_header client-ip`
	
	`add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' del_x_forwarded_for`
	
	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip`
	
	`add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'`
	
	`add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' insert_ns_client_header`
	
	`bind rewrite global check_x_forwarded_for_policy 100 200`
	
	`bind rewrite global check_client_ip_policy 200 300`
	
	`bind rewrite global insert_ns_client_policy 300 END`
	
	`set rewrite action insert_ns_client_header -stringBuilderExpr 'HTTP.REQ.HEADER("x-forwarded-for").VALUE(0) + " " + HTTP.REQ.HEADER("client-ip").VALUE(0) + " " + CLIENT.IP.SRC' -bypassSafetyCheck YES`

Tutorial Examples of Default Syntax Policies for Rewrite

With the rewrite feature, you can modify any part of an HTTP header, and, for responses, you can modify the HTTP body. You can use this feature to accomplish a number of useful tasks, such as removing unnecessary HTTP headers, masking internal URLs, redirecting Web pages, and redirecting queries or keywords.

In the examples listed in the table of contentes on the left side of your screen, you first create a rewrite action and a rewrite policy. Then you bind the policy globally.

Redirecting an External URL to an Internal URL

This example describes how to create a rewrite action and rewrite policy that redirects an external URL to an internal URL. You create an action, called act_external_to_internal, that performs the rewrite. Then you create a policy called pol_external_to_internal.

To redirect an external URL to an internal URL by using the command line

- To create the rewrite action, at the NetScaler command prompt, type:

  add rewrite action' act_external_to_internal REPLACE 'http.req.hostname.server' '"host_name_of_internal_Web_server"

- To create the rewrite policy, at the NetScaler command prompt, type:

  add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("host_name_of_external_Web_server")' act_external_to_internal

- Bind the policy globally.

To redirect an external URL to an internal URL by using the configuration utility

1. In the navigation pane, expand **Rewrite**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Rewrite Action** dialog box, enter the name **act_external_to_internal**.
4. To replace the HTTP server host name with the internal server name, choose **Replace** from the **Type** list box.
5. In the **Header Name** field, type **Host**.
6. In the **String expression for replacement** text field, type the internal host name of your Web server.
7. Click **Create** and then click **Close**.
8. In the navigation pane, click **Policies**.
9. In the details pane, click **Add**.
10. In the **Name** field, type **pol_external_to_internal**. This policy will detect connections to the Web server.
11. In the **Action** drop-down menu, choose the action **act_external_to_internal**.
12. In the **Expression** editor, construct the following expression:

    HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")

13. Bind your new policy globally.

Redirecting a Query

This example describes how to create a rewrite action and rewrite policy that redirects a query to the proper URL. The example assumes that the request contains a Host header set to **`www.example.com`** and a GET method with the **`string /query.cgi?server=5`**. The redirect extracts the domain name from the host header and the number from the query string, and redirects the user's query to the server **`Web5.example.com`**, where the rest of the user's query is processed.

Note: Although the following commands appears on multiple lines, you should enter them on a single line without line breaks.

To redirect a query to the appropriate URL using the command line

- To create a rewrite action named **act_redirect_query** that replaces the HTTP server host name with the internal server name, type:

  add rewrite action act_redirect_query REPLACE q#http.req.header("Host").before_str(".example.com")' '"Web" + http.req.url.query.value("server")#

- To create a rewrite policy named **pol_redirect_query**, type the following commands at the NetScaler command prompt.. This policy detects connections, to the Web server, that contain a query string. Do not apply this policy to connections that do not contain a query string:

  add rewrite policy pol_redirect_query q#http.req.header("Host").eq("www.example.com") && http.req.url.contains("?")' act_redirect_query#

- Bind your new policy globally.

Because this rewrite policy is highly specific and should be run before any other rewrite policies, it is advisable to assign it a high priority. If you assign it a priority of 1, it will be evaluated first.

Redirecting HTTP to HTTPS

This example describes how to rewrite Web server responses to find all URLs that begin with the string “http” and replace that string with “https.” You can use this to avoid having to update Web pages after moving a server from HTTP to HTTPS.

To redirect HTTP URLs to HTTPS by using the command line

- To create a rewrite action named **act_replace_http_with_https** that replaces all instances of the string “http” with the string “https,” at the NetScaler command prompt, type:

  add rewrite action act_replace_http_with_https replace_all 'http.res.body(100)' '"https"' -pattern http

- To create a rewrite policy named **pol_replace_http_with_https** that detects connections to the Web server, at the NetScaler command prompt, type:

  add rewrite policy pol_replace_http_with_https TRUE replace_https NOREWRITE

- Bind your new policy globally.

Removing Unwanted Headers

This example explains how to use a Rewrite policy to remove unwanted headers. Specifically, the example shows how to remove the following headers:

- **Accept Encoding header.** Removing the Accept Encoding header from HTTP responses prevents compression of the response.
- **Content Location header.** Removing the Content Location header from HTTP responses prevents your server from providing a hacker with information that might allow a security breach.

To delete headers from HTTP responses, you create a rewrite action and a rewrite policy, and you bind the policy globally.

To create the appropriate Rewrite action by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to either remove the Accept Encoding header and prevent response compression or remove the Content Location header:

- add rewrite action "act_remove-ae" delete_http_header "Accept-Encoding"
- add rewrite action "act_remove-cl" delete_http_header "Content-Location"

To create the appropriate Rewrite policy by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands to remove either the Accept Encoding header or the Content Location header:

- add rewrite policy "pol_remove-ae" true "act_remove-ae"
- add rewrite policy "pol_remove-cl" true "act_remove-cl"

To bind the policy globally by using the NetScaler command line

At the NetScaler command prompt, type one of the following commands, as appropriate, to globally bind the policy that you have created:

- bind rewrite global pol_remove_ae 100
- bind rewrite global pol_remove_cl 200

Reducing Web Server Redirects

This example explains how to use a Rewrite policy to modify connections to your home page and other URLs that end with a forward slash (/) to the default index page for your server, preventing redirects and reducing load on your server.

To modify directory-level HTTP requests to include the default home page by using the command line

- To create a Rewrite action named **action-default-homepage** that modifies URLs that end in a forward slash to include the default home page index.html, type:

  add rewrite action "action-default-homepage" replace q#http.req.url.path "/" "/index.html"#

- To create a Rewrite policy named **policy-default-homepage** that detects connections to your home page and applies your new action, type:

  add rewrite policy "policy-default-homepage" q#http.req.url.path.EQ("/") "action-default-homepage"#

- Globally bind your new policy to put it into effect.

Masking the Server Header

This example explains how to use a Rewrite policy to mask the information in the Server header in HTTP responses from your Web server. That header contains information that hackers can use to compromise your Web site. While masking the header will not prevent a skilled hacker from finding out information about your server, it will make hacking your Web server more difficult and encourage hackers to choose less well protected targets.

To mask the Server header in responses from the command line

1. To create a Rewrite action named **act_mask-server** that replaces the contents of the Server header with an uninformative string, type:

   add rewrite action "act_mask-server" replace "http.RES.HEADER(\"Server\")" "\"Web Server 1.0\""

2. To create a Rewrite policy named **pol_mask-server** that detects all connections, type:

   add rewrite policy "pol_mask-server" true "act_mask-server"

3. Globally bind your new policy to put it into effect.

Tutorial Examples of Classic Policies

The subtopics listed in the table of contents on the left side of your screen describe useful examples of classic policy configuration for certain NetScaler features such as Access Gateway, Application Firewall, and SSL.

Access Gateway Policy to Check for a Valid Client Certificate

The following policies enable the NetScaler to ensure that a client presents a valid certificate before establishing a connection to a company's SSL VPN.

To check for a valid client certificate by using the NetScaler command line

- At a NetScaler command prompt, create an Access Gateway profile named **act_current_client_cert** that requires that users have a current client certificate to establish an SSL connection with the Access Gateway or NetScaler.

  add ssl action act_current_client_cert-clientAuth DOCLIENTAUTH -clientCert ENABLED -certHeader "header_of_client_certificate_issued_by_your_company" -clientCertNotBefore ENABLED -certNotBeforeHeader "Mon, 01 Jan 2007 00:00:00 GMT"

- To create an SSL policy named **client_cert_policy** that detects connections to the Web server that contain a query string, type:

  add ssl policy client_cert_policy 'REQ.SSL.CLIENT.CERT.VALIDFROM >= "Mon, 01 Jan 2008 00:00:00 GMT"' act_block_ssl

- Globally bind your new policy to put it into effect.

Because this SSL policy should apply to any user's SSL connection unless a more specific SSL policy applies, you may want to assign a large priority value. For example, if you assign it a priority of one thousand (1000), that should ensure that other SSL policies are evaluated first, meaning that this policy will apply only to connections that do not match more specific policy criteria.

Application Firewall Policy to Protect a Shopping Cart Application

Shopping cart applications handle sensitive customer information, for example, credit card numbers and expiration dates, and they access back-end database servers. Many shopping cart applications also use legacy CGI scripts, which can contain security flaws that were unknown at the time they were written, but are now known to hackers and identity thieves.

A shopping cart application is particularly vulnerable to the following attacks:

- **Cookie tampering.** If a shopping cart application uses cookies, and does not perform the appropriate checks on the cookies that users return to the application, an attacker could modify a cookie and gain access to the shopping cart application under another user's credentials. Once logged on as that user, the attacker could obtain sensitive private information about the legitimate user or place orders using the legitimate user’s account.

- **SQL injection.** A shopping cart application normally accesses a back-end database server. Unless the application performs the appropriate safety checks on the data users return in the form fields of its Web forms before it passes that information on to the SQL database, an attacker can use a Web form to inject unauthorized SQL commands into the database server. Attackers normally use this type of attack to obtain sensitive private information from the database or modify information in the database.

The following configuration will protect a shopping cart application against these and other attacks.

To protect a shopping cart application by using the configuration utility

1. In the navigation pane, expand **Application Firewall**, click **Profiles**, and then click **Add**.
2. In the **Create Application Firewall Profile** dialog box, in the **Profile Name** field, enter **shopping_cart**.
3. In the **Profile Type** drop-down list, select **Web Application**.
4. In the **Configure** Select **Advanced** defaults.
5. Click **Create** and then click **Close**.
6. In the details view, double-click the new profile.
7. In the **Configure Web Application Profile** dialog box, configure your new profile as described below:

   a. Click the **Checks** tab, double-click the **Start URL** check, and in the **Modify Start URL Check** dialog box, click the **General** tab and disable blocking, and enable learning, logging, statistics, and URL closure. Click **OK** and then click **Close**.

   Note that if you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

   set appfw profile shopping_cart -startURLAction LEARN LOG STATS -startURLClosure ON

   b. For the **Cookie Consistency** check and **Form Field Consistency** checks, disable blocking, and enable learning, logging, statistics, using a similar method to the **Modify Start URL Check** configuration.

   If you are using the command line, you configure these settings by typing the following commands:

   set appfw profile shopping_cart -cookieConsistencyAction LEARN LOG STATS

   set appfw profile shopping_cart -fieldConsistencyAction LEARN LOG STATS

   c. For the **SQL Injection** check, disable blocking, and enable learning, logging, statistics, and transformation of special characters in the **Modify SQL Injection Check** dialog box, **General** tab, **Check Actions** section.

   If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

   set appfw profile shopping_cart -SQLInjectionAction LEARN LOG STATS -SQLInjectionTransformSpecialChars ON

d. For the **Credit Card** check, disable blocking; enable logging, statistics, and masking of credit card numbers; and enable protection for those credit cards you accept as forms of payment.

- If you are using the configuration utility, you configure blocking, logging, statistics, and masking (or *x-out*) in the Modify Credit Card Check dialog box, General tab, Check Actions section. You configure protection for specific credit cards in the Settings tab of the same dialog box.

- If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

set appfw profile shopping_cart -creditCardAction LOG STATS -creditCardXOut ON -creditCard <name> [<name>...]

For <name> you substitute the name of the credit card you want to protect. For Visa, you substitute **VISA**. For Master Card, you substitute **MasterCard**. For American Express, you substitute **Amex**. For Discover, you substitute **Discover**. For Diners Club, you substitute **DinersClub**. For JCB, you substitute **JCB**.

8. Create a policy named **shopping_cart** that detects connections to your shopping cart application and applies the **shopping_cart** profile to those connections.

To detect connections to the shopping cart, you examine the URL of incoming connections. If you host your shopping cart application on a separate host (a wise measure for security and other reasons), you can simply look for the presence of that host in the URL. If you host your shopping cart in a directory on a host that handles other traffic, as well, you must determine that the connection is going to the appropriate directory and/or HTML page.

The process for detecting either of these is the same; you create a policy based on the following expression, and substitute the proper host or URL for <string>.

REQ.HTTP.HEADER URL CONTAINS <string>

- If you are using the configuration utility, you navigate to the Application Firewall Policies page, click the Add... button to add a new policy, and follow the policy creation process described in “To create a policy with classic expressions using the configuration utility” beginning on page 201 and following.

- If you are using the command line, you type the following command at the prompt and press Enter:

  add appfw policy shopping_cart "REQ.HTTP.HEADER URL CONTAINS <string>" shopping_cart

9. Globally bind your new policy to put it into effect.

Because you want to ensure that this policy will match all connections to the shopping cart, and not be preempted by another more general policy, you should assign a high priority to it. If you assign one (1) as the priority, no other policy can preempt this one.

Application Firewall Policy to Protect Scripted Web Pages

Web pages with embedded scripts, especially legacy Javascripts, often violate the “same origin rule,” which does not allow scripts to access or modify content on any server but the server where they are located. This security vulnerability is called *cross-site scripting*. The Application Firewall Cross-Site Scripting rule normally filters out requests that contain cross-site scripting.

Unfortunately, this can cause Web pages with older Javascripts to stop functioning, even when your system administrator has checked those scripts and knows that they are safe. The example below explains how to configure the Application Firewall to allow cross-site scripting in Web pages from trusted sources without disabling this important filter for the rest of your Web sites.

To protect Web pages with cross-site scripting by using the NetScaler command line

- At the NetScaler command line, to create an advanced profile, type:

  add appfw profile pr_xssokay -defaults advanced

- To configure the profile, type:

  set appfw profile pr_xssokay -startURLAction NONE -startURLClosure OFF -cookieConsistencyAction LEARN LOG STATS -fieldConsistencyAction LEARN LOG STATS -crossSiteScriptingAction LEARN LOG STATS$"

- Create a policy that detects connections to your scripted Web pages and applies the **pr_xssokay** profile, type:

  add appfw policy pol_xssokay "REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$ || REQ.HTTP.HEADER URL CONTAINS ^\.js$" pr_xssokay

- Globally bind the policy.

To protect Web pages with cross-site scripting by using the configuration utility

1. In the navigation pane, expand **Application Firewall**, and then click **Profiles**.
2. In the details view, click **Add**.
3. In the **Create Application Firewall Profile** dialog box, create a Web Application profile with advanced defaults and name it **pr_xssokay**. Click **Create** and then click **Close**.
4. In the details view, click the profile, click **Open**, and in the **Configure Web Application Profile** dialog box, configure the **pr_xssokay** profile as shown below.

   Start URL Check: Clear all actions.

   - Cookie Consistency Check: Disable blocking.
   - Form Field Consistency Check: Disable blocking.
   - Cross-Site Scripting Check: Disable blocking.

   This should prevent blocking of legitimate requests involving Web pages with cross-site scripting that you know are nonetheless safe.
5. Click **Policies**, and then click **Add**.
6. In the **Create Application Firewall Policy** dialog box, create a policy that detects connections to your scripted Web pages and applies the **pr_xssokay** profile:

   - Policy name: **pol_xssokay**
   - Associated profile: **pr_xssokay**

   Policy expression: `"REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$ || REQ.HTTP.HEADER URL CONTAINS ^\.js$"`
7. Globally bind your new policy to put it into effect.

DNS Policy to Drop Packets from Specific IPs

The following example describes how to create a DNS action and DNS policy that detects connections from unwanted IPs or networks, such as those used in a DDOS attack, and drops all packets from those locations. The example shows networks within the IANA reserved IP block `192.168.0.0/16`. A hostile network will normally be on publicly routable IPs.

To drop packets from specific IPs by using the NetScaler command line

- To create a DNS policy named **pol_ddos_drop** that detects connections from hostile networks and drops those packets, type:

  add dns policy pol_ddos_drop 'client.ip.src.in_subnet(192.168.253.128/25) || client.ip.src.in_subnet(192.168.254.32/27)' -drop YES'

  For the example networks in the `192.168.0.0/16` range, you substitute the IP and netmask in ***###.###.###.###/##*** format of each network you want to block. You can include as many networks as you want, separating each `CLIENT.IP.SRC.IN_SUBNET(###.###.###.###./##)` command with the OR operator.

- Globally bind your new policy to put it into effect.

SSL Policy to Require Valid Client Certificates

The following example shows an SSL policy that checks the user's client certificate validity before initiating an SSL connection with a client.

To block connections from users with expired client certificates

- Log on to the NetScaler command line.

  If you are using the GUI, navigate to the **SSL Policies** page, then in the **Data** area, click the **Actions** tab.

- Create an SSL action named **act_current_client_cert** that requires that users have a current client certificate to establish an SSL connection with the NetScaler.

  add ssl action act_current_client_cert-clientAuth DOCLIENTAUTH -clientCert ENABLED -certHeader "clientCertificateHeader" -clientCertNotBefore ENABLED -certNotBeforeHeader "Mon, 01 Jan 2007 00:00:00 GMT"

- Create an SSL policy named **pol_current_client_cert** that detects connections to the Web server that contain a query string.

  add ssl policy pol_current_ client_cert 'REQ.SSL.CLIENT.CERT.VALIDFROM >= "Mon, 01 Jan 2007 00:00:00 GMT"' act_block_ssl

- Bind your new policy globally.

  Because this SSL policy should apply to any user’s SSL connection unless a more specific SSL policy applies, you may want to assign it a low priority. If you assign it a priority of one thousand (1000), that should ensure that other SSL policies are evaluated first, meaning that this policy will apply only to connections that do not match more specific policy criteria.

Migration of Apache mod_rewrite Rules to the Default Syntax

The Apache HTTP Server provides an engine known as mod_rewrite for rewriting HTTP request URLs. If you migrate the mod_rewrite rules from Apache to the NetScaler, you boost back-end server performance. In addition, because the NetScaler typically load balances multiple (sometimes thousands of) Web servers, after migrating the rules to the NetScaler you will have a single point of control for these rules.

The subtopic listed in the table of contents on the left side of your screen provide examples of mod_rewrite functions, and translations of these functions into Rewrite and Responder policies on the NetScaler.

Converting URL Variations into Canonical URLs

On some Web servers you can have multiple URLs for a resource. Although the canonical URLs should be used and distributed, other URLs can exist as shortcuts or internal URLs. You can make sure that users see the canonical URL regardless of the URL used to make an initial request.

In the following examples, the URL /~user is converted to /u/user.

Apache mod_rewrite solution for converting a URL

```
RewriteRule   ^/~([^/]+)/?(.*)    /u/$1/$2[R]
```

NetScaler solution for converting a URL

```
add responder action act1 redirect '"/u/"+HTTP.REQ.URL.AFTER_STR("/~")' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.STARTSWITH("/~") && HTTP.REQ.URL.LENGTH.GT(2)' act1
bind responder global pol1 100
```

Converting Host Name Variations to Canonical Host Names

You can enforce the use of a particular host name for reaching a site. For example, you can enforce the use of www.example.com instead of example.com.

Apache mod_rewrite solution for enforcing a particular host name for sites running on a port other than 80

```
RewriteCond %{HTTP_HOST}   !^www.example.com
RewriteCond %{HTTP_HOST}   !^$
RewriteCond %{SERVER_PORT} !^80$
RewriteRule ^/(.*)         http://www.example.com:%{SERVER_PORT}/$1 [L,R]
```

Apache mod_rewrite solution for enforcing a particular host name for sites running on port 80

```
RewriteCond %{HTTP_HOST}   !^www.example.com
RewriteCond %{HTTP_HOST}   !^$
RewriteRule ^/(.*)         http://www.example.com/$1 [L,R]
```

NetScaler solution for enforcing a particular host name for sites running on a port other than 80

```
add responder action act1 redirect '"http://www.example.com:"+CLIENT.TCP.DSTPORT+HTTP.REQ.URL' -byp
add responder policy pol1 '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com")&&!HTTP.REQ.HOSTNAME.
bind responder global pol1 100 END
```

NetScaler solution for enforcing a particular host name for sites running on port 80

```
add responder action act1 redirect '"http://www.example.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1  '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com")&&!HTTP.REQ.HOSTNAME
bind responder global  pol1 100 END
```

Moving a Document Root

Usually the document root of a Web server is based on the URL “/”. However, the document root can be any directory. You can redirect traffic to the document root if it changes from the top-level “/” directory to another directory.

In the following examples, you change the document root from / to /e/www. The first two examples simply replace one string with another. The third example is more universal because, along with replacing the root directory, it preserves the rest of the URL (the path and query string), for example, redirecting /example/file.html to /e/www/example/file.html.

Apache mod_rewrite solution for moving the document root

```
RewriteEngine on
RewriteRule   ^/$  /e/www/  [R]
```

NetScaler solution for moving the document root

```
add responder action act1 redirect '"/e/www/"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.EQ("/")' act1
bind responder global pol1 100
```

NetScaler solution for moving the document root and appending path information to the request

```
add responder action act1 redirect '"/e/www"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.URL.STARTSWITH("/e/www/")' act1
bind responder global pol1 100 END
```

Moving Home Directories to a New Web Server

You may want to redirect requests that are sent to home directories on a Web server to a different Web server. For example, if a new Web server is replacing an old one over time, as you migrate home directories to the new location you need to redirect requests for the migrated home directories to the new Web server.

In the following examples, the host name for the new Web server is newserver.

Apache mod_rewrite solution for redirecting to another Web server

```
RewriteRule   ^/(.+)  http://newserver/$1     [R,L]
```

NetScaler solution for redirecting to another Web server (method 1)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.REGEX_MATCH(re#^/(.+)#)'    act1
bind responder global pol1 100 END
```

NetScaler solution for redirecting to another Web server (method 2)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.LENGTH.GT(1)' act1
bind responder global pol1 100 END
```

Working with Structured Home Directories

Typically, a site with thousands of users has a structured home directory layout. For example, each home directory may reside under a subdirectory that is named using the first character of the user name. For example, the home directory for jsmith (/~jsmith/anypath) might be /home/j/smith/.www/anypath, and the home directory for rvalveti (/~rvalveti/anypath) might be /home/r/rvalveti/.www/anypath.

The following examples redirect requests to the home directory.

Apache mod_rewrite solution for structured home directories

```
RewriteRule   ^/~(([a-z])[a-z0-9]+)(.*)  /home/$2/$1/.www$3
```

NetScaler solution for structured home directories

NetScaler solution for structured home directories

```
add rewrite action act1 replace 'HTTP.REQ.URL'  '"/home/"+ HTTP.REQ.URL.AFTER_STR("~").PREFIX(1)+"/"+ H
add rewrite policy pol1  'HTTP.REQ.URL.PATH.STARTSWITH("/~")' act1
bind rewrite global pol1 100
```

Redirecting Invalid URLs to Other Web Servers

If a URL is not valid, it should be redirected to another Web server. For example, you should redirect to another Web server if a file that is named in a URL does not exist on the server that is named in the URL.

On Apache, you can perform this check using mod_rewrite. On the NetScaler, an HTTP callout can check for a file on a server by running a script on the server. In the following NetScaler examples, a script named file_check.cgi processes the URL and uses this information to check for the presence of the target file on the server. The script returns TRUE or FALSE, and the NetScaler uses the value that the script returns to validate the policy.

In addition to performing the redirection, the NetScaler can add custom headers or, as in the second NetScaler example, it can add text in the response body.

Apache mod_rewrite solution for redirection if a URL is wrong

```
RewriteCond   /your/docroot/%{REQUEST_FILENAME} !-f
RewriteRule   ^(.+)      http://webserverB.com/$1 [R]
```

NetScaler solution for redirection if a URL is wrong (method 1)

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -R
add responder action act1 redirect '"http://webserverB.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.HTTP_CALLOUT(call)' act1
bind responder global pol1 100
```

NetScaler solution for redirection if a URL is wrong (method 2)

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -R
add responder  action act1 respondwith  '"HTTP/1.1 302 Moved Temporarily\r\nLocation: http://webserverB.
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.HTTP_CALLOUT(call)' act1
bind responder global pol1 100
```

Rewriting a URL Based on Time

You can rewrite a URL based on the time. The following examples change a request for example.html to example.day.html or example.night.html, depending on the time of day.

Apache mod_rewrite solution for rewriting a URL based on the time

```
RewriteCond   %{TIME_HOUR}%{TIME_MIN} >0700
RewriteCond   %{TIME_HOUR}%{TIME_MIN} <1900
RewriteRule   ^example\.html$ example.day.html [L]
RewriteRule   ^example\.html$ example.night.html
```

NetScaler solution for rewriting a URL based on the time

```
add rewrite action act1 insert_before 'HTTP.REQ.URL.PATH.SUFFIX(\'.\',0)' '"day."'
add rewrite action act2  insert_before 'HTTP.REQ.URL.PATH.SUFFIX(\'.\',0)' '"night."'
add rewrite  policy pol1 'SYS.TIME.WITHIN(LOCAL 07h 00m,LOCAL 18h 59m)' act1
add rewrite policy pol2 'true'  act2
bind rewrite global pol1 101
bind rewrite global pol2 102
```

Redirecting to a New File Name (Invisible to the User)

If you rename a Web page, you can continue to support the old URL for backward compatibility while preventing users from recognizing that the page was renamed.

In the first two of the following examples, the base directory is /~quux/. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for managing a file name change in a fixed location

```
RewriteEngine  on
RewriteBase    /~quux/
RewriteRule    ^foo\.html$  bar.html
```

NetScaler solution for managing a file name change in a fixed location

```
add rewrite action act1 replace 'HTTP.REQ.URL.AFTER_STR("/~quux").SUBSTR("foo.html")' '"bar.html"'
add rewrite policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/foo.html")' act1
bind rewrite global pol1 100
```

NetScaler solution for managing a file name change regardless of the base directory or query strings in the URL

```
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"bar.html"'
Add rewrite policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("foo.html")' act1
Bind rewrite global pol1 100
```

Redirecting to New File Name (User-Visible URL)

If you rename a Web page, you may want to continue to support the old URL for backward compatibility and allow users to see that the page was renamed by changing the URL that is displayed in the browser.

In the first two of the following examples, redirection occurs when the base directory is /~quux/. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for changing the file name and the URL displayed in the browser

```
RewriteEngine on
RewriteBase    /~quux/
RewriteRule    ^old\.html$ new.html  [R]
```

NetScaler solution for changing the file name and the URL displayed in the browser

```
add responder action act1 redirect 'HTTP.REQ.URL.BEFORE_STR("foo.html")+"new.html"' -bypassSafetyCheck
add responder policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/old.html")' act1
bind responder global pol1 100
```

NetScaler solution for changing the file name and the URL displayed in the browser regardless of the base directory or query strings in the URL

```
add responder action act1 redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("old.html")+"new.html"+HTTP.REQ.URL
add responder policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("old.html")' act1
bind responder global pol1 100
```

Accommodating Browser Dependent Content

To accommodate browser-specific limitations—at least for important top-level pages—it is sometimes necessary to set restrictions on the browser type and version. For example, you might want to set a maximum version for the latest Netscape variants, a minimum version for Lynx browsers, and an average feature version for all others.

The following examples act on the HTTP header "User-Agent", such that if this header begins with "Mozilla/3", the page MyPage.html is rewritten to MyPage.NS.html. If the browser is "Lynx" or "Mozilla" version 1 or 2, the URL becomes MyPage.20.html. All other browsers receive page MyPage.32.html.

Apache mod_rewrite solution for browser-specific settings

```
RewriteCond %{HTTP_USER_AGENT}  ^Mozilla/3.*
RewriteRule ^MyPage\.html$ MyPage.NS.html [L]
RewriteCond %{HTTP_USER_AGENT}  ^Lynx/.* [OR]
RewriteCond %{HTTP_USER_AGENT}  ^Mozilla/[12].*
RewriteRule ^MyPage\.html$ MyPage.20.html [L]
RewriteRule ^fMyPage\.html$ MyPage.32.html [L]
NetScaler solution for browser-specific settings
add patset pat1
bind patset pat1 Mozilla/1
bind Patset pat1 Mozilla/2
bind patset pat1 Lynx
bind Patset pat1 Mozilla/3
add rewrite action act1 insert_before 'HTTP.REQ.URL.SUFFIX' '"NS."'
add rewrite action act2 insert_before 'HTTP.REQ.URL.SUFFIX' '"20."'
add rewrite action act3 insert_before 'HTTP.REQ.URL.SUFFIX' '"32."'
add rewrite policy pol1 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX("pat1").EQ(4)' act1
add rewrite policy pol2 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX("pat1").BETWEEN(1,3)' act2
add rewrite policy pol3 '!HTTP.REQ.HEADER("User-Agent").STARTSWITH_ANY("pat1")' act3
bind rewrite global pol1 101 END
bind rewrite global pol2 102 END
bind rewrite global pol3 103 END
```

Blocking Access by Robots

You can block a robot from retrieving pages from a specific directory or a set of directories to ease up the traffic to and from these directories. You can restrict access based on the specific location or you can block requests based on information in User-Agent HTTP headers.

In the following examples, the Web location to be blocked is /~quux/foo/arc/, the IP addresses to be blocked are 123.45.67.8 and 123.45.67.9, and the robot's name is NameOfBadRobot.

Apache mod_rewrite solution for blocking a path and a User-Agent header

```
RewriteCond %{HTTP_USER_AGENT}   ^NameOfBadRobot.*
RewriteCond %{REMOTE_ADDR}       ^123\.45\.67\.[8-9]$
RewriteRule ^/~quux/foo/arc/.+   -   [F]
```

NetScaler solution for blocking a path and a User-Agent header

```
add responder action act1 respondwith "'HTTP/1.1 403 Forbidden\r\n\r\n'"
add responder policy pol1 'HTTP.REQ.HEADER("User_Agent").STARTSWITH("NameOfBadRobot")&&CLIENT.IP.S
bind responder global pol1 100
```

Blocking Access to Inline Images

If you find people frequently going to your server to copy inline graphics for their own use (and generating unnecessary traffic), you may want to restrict the browser's ability to send an HTTP Referer header.

In the following example, the graphics are located in http://www.quux-corp.de/~quux/.

Apache mod_rewrite solution for blocking access to an inline image

```
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://www.quux-corp.de/~quux/.*$
RewriteRule .*\.gif$ - [F]
```

NetScaler solution for blocking access to an inline image

```
add patset pat1
bind patset pat1 .gif
bind patset pat1 .jpeg
add responder action act1 respondwith "'HTTP/1.1 403 Forbidden\r\n\r\n'"
add responder policy pol1 '!HTTP.REQ.HEADER("Referer").EQ("") && !HTTP.REQ.HEADER("Referer").STARTSWI
bind responder global pol1 100
```

Creating Extensionless Links

To prevent users from knowing application or script details on the server side, you can hide file extensions from users. To do this, you may want to support extensionless links. You can achieve this behavior by using rewrite rules to add an extension to all requests, or to selectively add extensions to requests.

The first two of the following examples show adding an extension to all request URLs. In the last example, one of two file extensions is added. Note that in the last example, the mod_rewrite module can easily find the file extension because this module resides on the Web server. In contrast, the NetScaler must invoke an HTTP callout to check the extension of the requested file on the Web server. Based on the callout response, the NetScaler adds the .html or .php extension to the request URL.

Note: In the second NetScaler example, an HTTP callout is used to query a script named file_check.cgi hosted on the server. This script checks whether the argument that is provided in the callout is a valid file name.

Apache mod_rewrite solution for adding a .php extension to all requests

```
RewriteRule ^/?([a-z]+)$ $1.php [L]
```

NetScaler policy for adding a .php extension to all requests

```
add rewrite action act1 insert_after 'HTTP.REQ.URL' "".php""
add rewrite policy pol1 'HTTP.REQ.URL.PATH.REGEX_MATCH(re#^/([a-z]+)$#)' act1
bind rewrite global pol1 100
```

Apache mod_rewrite solution for adding either .html or .php extensions to requests

```
RewriteCond %{REQUEST_FILENAME}.php -f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.php [L]
RewriteCond %{REQUEST_FILENAME}.html –f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.html [L]
```

NetScaler policy for adding either .html or .php extensions to requests

```
add HTTPCallout Call_html
add HTTPCallout Call_php
set policy httpCallout Call_html -IPAddress 10.102.59.101 -port 80 -hostExpr ""10.102.59.101"" -returnType BO
set policy httpCallout Call_php -IPAddress 10.102.59.101 -port 80 -hostExpr ""10.102.59.101"" -returnType BO
add patset pat1
bind patset pat1 .html
bind patset pat1 .php
bind patset pat1 .asp
```

```
bind patset pat1 .cgi
add rewrite  action act1 insert_after 'HTTP.REQ.URL.PATH'  "".html""
add rewrite  action act2 insert_after "HTTP.REQ.URL.PATH"  "".php""
add rewrite policy pol1 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.HTTP_CALLOUT(Call_html)' act1
add rewrite policy pol2 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.HTTP_CALLOUT(Call_php)' act2
bind rewrite global pol1 100 END
bind rewrite global pol2 101 END
```

Redirecting a Working URI to a New Format

Suppose that you have a set of working URLs that resemble the following:

/index.php?id=nnnn

To change these URLs to /nnnn and make sure that search engines update their indexes to the new URI format, you need to do the following:

- Redirect the old URIs to the new ones so that search engines update their indexes.
- Rewrite the new URI back to the old one so that the index.php script runs correctly.

To accomplish this, you can insert marker code into the query string (making sure that the marker code is not seen by visitors), and then removing the marker code for the index.php script.

The following examples redirect from an old link to a new format only if a marker is not present in the query string. The link that uses the new format is re-written back to the old format, and a marker is added to the query string.

Apache mod_rewrite solution

```
RewriteCond %{QUERY_STRING} !marker
RewriteCond %{QUERY_STRING} id=([-a-zA-Z0-9_+]+)
RewriteRule ^/?index\.php$ %1? [R,L]
RewriteRule ^/?([-a-zA-Z0-9_+]+)$  index.php?marker&id=$1 [L]
NetScaler solution
add responder action act_redirect redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("index.php")+HTTP.REQ.URL.C
add responder policy pol_redirect '!HTTP.REQ.URL.QUERY.CONTAINS("marker")&& HTTP.REQ.URL.QUERY.VA
bind responder global pol_redirect 100 END
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"index.phpmarker&id="+HTTP.REQ.URL
add rewrite policy pol1 '!HTTP.REQ.URL.QUERY.CONTAINS("marker")'  act1
bind rewrite global pol1 100 END
```

Ensuring That a Secure Server Is Used for Selected Pages

To make sure that only secure servers are used for selected Web pages, you can use the following Apache mod_rewrite code or NetScaler Responder policies.

Apache mod_rewrite solution

```
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^/?(page1|page2|page3|page4|page5)$  https://www.example.com/%1 [R,L]
```

NetScaler solution using regular expressions

```
add responder action res_redirect redirect  '"https://www.example.com"+HTTP.REQ.URL' -bypassSafetyChec
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.REQ.URL.REGEX_MATCH(re/page[
bind responder global pol_redirect 100 END
```

NetScaler solution using pattern sets

```
add patset pat1
bind patset pat1 page1
bind patset pat1 page2
bind patset pat1 page3
bind patset pat1 page4
bind patset pat1 page5
add responder action res_redirect redirect  '"https://www.example.com"+HTTP.REQ.URL' -bypassSafetyChec
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.REQ.URL.CONTAINS_ANY("pat1")'
bind responder global pol_redirect 100 END
```