

Configuration Guide for BIG-IP® Link Controller™

version 11.0

MAN-0308-02



Product Version

This manual applies to product version 11.0 of the BIG-IP® Link Controller™.

Publication Data

This manual was published on August 11, 2011.

Legal Notices

Copyright

Copyright © 2002-2011, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

3DNS, Access Policy Manager, Acopia, Acopia Networks, Advanced Client Authentication, Advanced Routing, APM, Application Security Manager, ARX, AskF5, ASM, BIG-IP, Cloud Extender, CloudFucious, CMP, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, EM, Enterprise Manager, F5, F5 [DESIGN], F5 Management Pack, F5 Networks, F5 World, Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iApps, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, IT agility. Your way., L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, Message Security Module, MSM, Netcelera, OneConnect, Packet Velocity, Protocol Security Module, PSM, Real Traffic Policy Builder, Scale^N, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TMOS, Traffic Management Operating System, TrafficShield, Transparent Data Reduction, VIPRION, vCMP, WA, WAN Optimization Manager, WANJet, WebAccelerator, WOM, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by U.S. Patent 7,945,678. This list is believed to be current as of August 11, 2011.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This class A digital apparatus complies with Canadian I CES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Gabriel Forté.

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, <http://www.and.com>.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems.

"Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD," "FreeBSD," "Mach" (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at <http://www.perl.com>.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product contains software based on *oprofile*, which is protected under the GNU Public License.

This product includes *RRDtool* software developed by Tobi Oetiker (<http://www.rrdtool.com/index.html>) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation <<http://www.apache.org/>>.

This product includes *Hypersonic SQL*.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (<http://www.nominum.com>).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes the GeoPoint Database developed by Quova, Inc. and its contributors.

This product includes software developed by Balazs Scheidler <bazsi@balabit.hu>, which is protected under the GNU Public License.

This product includes software developed by NLnet Labs and its contributors.

This product includes software written by Steffen Beyer and licensed under the Perl Artistic License and the GPL.

This product includes software written by Makamaka Hannyaharamitu © 2007-2008.



Table of Contents

I

Introducing the Link Controller

Introducing the BIG-IP system	1-1
Introducing the Link Controller	1-1
Understanding link load balancing	1-2

2

Essential Configuration Tasks

Introducing essential configuration tasks	2-1
Creating the default gateway pool	2-2
Implementing the default gateway pool	2-2
Adding links	2-3
Adding listeners	2-4
Adding pools	2-5
Adding virtual servers	2-6
Adding wide IPs	2-7
Configuring address translation	2-8

3

Defining Links

Introducing links	3-1
Adding links	3-2
Adding health monitors	3-3
Monitoring the Link Controller	3-3
Monitoring with SNMP	3-4
Determining availability requirements	3-4
Setting bandwidth restrictions	3-5
Configuring link weighting and billing properties	3-6
Configuring duplex billing for each link	3-7
Configuring price weighting for each link	3-7
Configuring link ratio weighting for each link	3-8

4

Working with Listeners

Introducing listeners	4-1
Creating a listener for local resolution	4-2
Configuring listeners for traffic forwarding	4-3
Configuring a wildcard listener	4-4
Modifying listeners	4-4
Deleting listeners	4-5
Using listeners with VLANs	4-6
Setting up a listener for all VLANs	4-6
Enabling a listener for specific VLANs	4-7
Disabling a listener for specific VLANs	4-7

5

Configuring Load Balancing Pools

Introducing load balancing pools	5-1
What is a load balancing pool?	5-1
Features of a load balancing pool	5-1
Creating and modifying load balancing pools	5-2
Creating and implementing a load balancing pool	5-2

Modifying a load balancing pool	5-3
Modifying pool membership	5-3
Configuring pool settings	5-6
Specifying a pool name	5-7
Associating health monitors with a pool	5-8
Specifying the availability requirements	5-9
Allowing SNATs and NATs	5-9
Specifying action when a service becomes unavailable	5-9
Configuring a slow ramp time	5-9
Configuring the Type of Service (ToS) level	5-11
Configuring the Quality of Service (QoS) level	5-11
Specifying the load balancing method	5-12
Specifying priority-based member activation	5-15
Specifying pool members	5-16
Configuring pool member settings	5-16
Specifying a ratio weight for a pool member	5-17
Specifying priority-based member activation	5-17
Specifying a connection limit	5-18
Selecting an explicit monitor association	5-18
Creating an explicit monitor association for a pool member	5-18
Specifying an availability requirement	5-19
Managing pools and pool members	5-20
Managing pools	5-20
Managing pool members	5-23
Removing monitor associations	5-27
Viewing pool and pool member statistics	5-28

6

Configuring Virtual Servers

Introducing virtual servers and virtual addresses	6-1
What is a virtual server?	6-1
What is a virtual address?	6-2
Understanding virtual server types	6-3
Host virtual servers	6-3
Network virtual servers	6-3
Creating a virtual server	6-6
Understanding virtual server and virtual address settings	6-9
Configuring virtual server settings	6-9
Configuring virtual address settings	6-13
Managing virtual servers and virtual addresses	6-15
Viewing or modifying a virtual server configuration	6-15
Viewing or modifying a virtual address configuration	6-17
Understanding virtual server and virtual address status	6-18
Enabling or disabling a virtual server or virtual address	6-19
Deleting a virtual server or virtual address	6-20

7

Defining Wide IPs

Introducing wide IPs	7-1
Configuring wide IPs	7-1
Using wildcard characters in wide IP names	7-2
Modifying a wide IP	7-3

8

Configuring Monitors

Introducing monitors	8-1
Summary of monitor types	8-2
Summary of status types	8-3
Understanding pre-configured and custom monitors	8-4
Creating a custom monitor	8-6
Configuring monitor settings	8-7
Simple monitors	8-7
Extended Content Verification (ECV) monitors	8-10
FTP monitor	8-12
Special configuration considerations	8-13
Setting destinations	8-13
Using transparent and reverse modes	8-14
Associating monitors with resources	8-15
Types of monitor associations	8-16
Managing monitors	8-17

9

Inbound Load Balancing

Introducing inbound load balancing	9-1
Understanding inbound load balancing on the Link Controller	9-1
Using static load balancing modes	9-3
Drop Packet mode	9-3
Fallback IP mode	9-3
Global Availability mode	9-4
Ratio mode	9-4
Round Robin mode	9-4
Static Persist mode	9-4
Topology mode	9-5
Using dynamic load balancing modes	9-6
Types of dynamic load balancing modes	9-6
Implementing the Quality of Service load balancing mode	9-8
Using the Dynamic Ratio option	9-11
Configuring inbound load balancing	9-13
Changing the load balancing methods	9-13
Using the Round Robin LDNS wide IP attribute	9-13
Adjusting the QOS coefficients	9-14

10

Working with Topologies

Introducing topologies	10-1
Understanding topologies	10-1
Implementing topologies	10-2
Setting up and removing topology records	10-3
Removing topology records	10-4
Using topology load balancing in a wide IP	10-5
Understanding user-defined regions	10-6
Other load balancing options for topologies	10-7

11

Synchronizing Link Controllers

Introducing synchronization	11-1
Defining NTP servers	11-2
Activating synchronization	11-3
Controlling file synchronization	11-3
Deactivating file synchronization	11-4
Creating synchronization groups	11-4
Running the gtm_add script	11-5
Synchronizing Link Controller and Global Traffic Manager systems	11-6

12

Viewing Statistics

Introducing statistics	12-1
Accessing statistics	12-1
Understanding the types of statistics	12-3
Wide IP statistics	12-3
Wide IP member statistics	12-5
Link statistics	12-6
Paths statistics	12-7
Local DNS statistics	12-8

13

Understanding Profiles

Introducing profiles	13-1
Profile types	13-1
Default profiles	13-2
Custom and parent profiles	13-3
Summarizing profiles	13-4
Creating and modifying profiles	13-5
Using a default profile as is	13-5
Modifying a default profile	13-5
Creating a custom profile	13-6
Modifying a custom profile	13-8
Viewing and deleting profiles	13-9
Viewing a list of profiles	13-9
Deleting a profile	13-10
Implementing a profile	13-10
For more information	13-12

14

Managing HTTP and FTP Traffic

Introducing HTTP and FTP traffic management	14-1
Configuring HTTP standard profile settings	14-2
Understanding HTTP profile settings	14-2
Configuring FTP profile settings	14-6
Specifying a profile name	14-6
Specifying a parent profile	14-6
Specifying a Translate Extended value	14-6
Specifying a data port	14-7

15

Enabling Session Persistence

Introducing session persistence	15-1
Configuring a persistence profile	15-1
Enabling session persistence through iRules	15-2
Persistence types and their profiles	15-2
Types of persistence	15-2
Understanding criteria for session persistence	15-2
Destination address affinity persistence	15-4
Source address affinity persistence	15-5

16

Managing Protocol Profiles

Introducing protocol profiles	16-1
Configuring a Fast L4 profile	16-2
Understanding Fast L4 profile settings	16-2
Configuring a TCP profile	16-4
Understanding TCP profile settings	16-5
Configuring a UDP profile	16-9

17

Using the Statistics Profile

Introducing the Statistics profile	17-1
Configuring a Statistics profile	17-2

18

Writing iRules

Introducing iRules	18-1
What is an iRule?	18-1
Basic iRule elements	18-2
Specifying traffic destinations and address translations	18-3
Creating iRules	18-5
Controlling iRule evaluation	18-6
Configuration prerequisites	18-6
Specifying events	18-6
Using iRule commands	18-9
Statement commands	18-9
Query and manipulation commands	18-9
Utility commands	18-9
Working with profiles	18-10
Reading profile settings	18-10
Overriding profile settings	18-10
Enabling session persistence with iRules	18-11
Creating, managing, and using data groups	18-11
Using the matchclass command	18-11
Creating data groups	18-12
Storage options	18-14
Displaying data group properties	18-16
Managing data group members	18-16

19

Configuring SNATs and NATs

Introducing secure network address translation	19-1
How does a SNAT work?	19-2
Mapping original IP addresses to translation addresses	19-2
Creating a SNAT pool	19-4
Implementing a SNAT	19-5
Creating a standard SNAT	19-6
Creating an intelligent SNAT	19-9
Assigning a SNAT pool directly to a virtual server	19-9
Implementing a NAT	19-10
Additional restrictions	19-11
Managing SNATs and NATs	19-12
Viewing or modifying SNATs, NATs, and SNAT pools	19-12
Defining and viewing translation addresses	19-13
Deleting SNATs, NATs, SNAT pools, and translation addresses	19-13
Enabling or disabling SNATs or NATs for a load balancing pool	19-14
Enabling or disabling SNAT translation addresses	19-14
SNAT examples	19-15
Example 1 - Establishing a standard SNAT that uses a SNAT pool	19-15
Example 2 - Establishing an intelligent SNAT	19-16

20

Configuring Nodes

Introducing nodes	20-1
Creating and modifying nodes	20-2
Configuring node settings	20-3
Specifying an address for a node	20-4
Specifying a node name	20-4
Assigning health monitors	20-4
Specifying the availability requirement	20-5
Specifying a ratio weight	20-6
Setting a connection limit	20-6
Managing nodes	20-7
Viewing a list of nodes	20-7
Viewing node properties	20-7
Understanding node status	20-8
Enabling or disabling a node	20-9
Deleting a node	20-10
Removing monitor associations	20-10

21

Configuring Rate Shaping

Introducing rate shaping	21-1
Creating and implementing rate classes	21-2
Configuring rate class settings	21-3
Specifying a name	21-4
Specifying a base rate	21-4
Specifying a ceiling rate	21-4
Specifying a burst size	21-4
Specifying direction	21-7
Specifying a parent class	21-7
Specifying a queue method	21-8
Managing rate classes	21-9

A

Additional Monitor Considerations

Implementing monitors for Dynamic Ratio load balancing	A-1
Implementing a Real Server monitor	A-1
Implementing a WMI monitor	A-3
Implementing an SNMP DCA or SNMP DCA Base monitor	A-5
Implementing an MSSQL monitor	A-6

Glossary

Index



I

Introducing the Link Controller

- Introducing the BIG-IP system
- Introducing the Link Controller

Introducing the BIG-IP system

F5 Networks' BIG-IP® system is a port-based, multilayer switch that supports virtual local area network (VLAN) technology. Because hosts within a VLAN can communicate at the data-link layer (Layer 2), a BIG-IP system reduces the need for routers and IP routing on the network. This in turn reduces equipment costs and boosts overall network performance. At the same time, the BIG-IP system's multilayer capabilities enable the system to process traffic at other OSI layers. The BIG-IP system can perform IP routing at Layer 3, as well as manage and secure TCP, UDP, and other application traffic at Layer 4 through 7.

Introducing the Link Controller

The BIG-IP® Link Controller™ system is a dedicated IP application switch that manages traffic to and from a site across multiple links, regardless of connection type or provider. The Link Controller system provides granular traffic control for Internet gateways, allowing you to define how traffic is distributed across links in a way that meets your business priorities. The Link Controller system also transparently monitors the availability and health of links to optimally direct traffic across the best available link.

The Link Controller system runs on the Traffic Management Operating System® commonly referred to as TMOS®. Before you begin configuring Link Controller, run the Setup utility to configure basic network elements such as self IP addresses, interfaces, and VLANs. After running the Setup utility, you can use this guide to configure specific implementations.

The Link Controller system includes the following features:

- ◆ Dynamic load balancing, based on the following link attributes:
 - Total available bandwidth of the link
 - The costs of purchased incremental bandwidth segments
 - Inbound link capacity and resource limits
 - Outbound link capacity and resource limits
- ◆ Router monitoring, to ensure high availability and continuous uptime

Understanding link load balancing

Link load balancing is defined as managing traffic across multiple Internet or wide-area network (WAN) gateways. Link load balancing ensures high availability in the network, and improves the performance of a web site or data center. Link load balancing provides a method by which an organization can reliably manage a multi-homed network. A **multi-homed network** is composed of one or more data centers that have more than one link to the Internet.

As enterprises increase their reliance on the Internet for delivering mission-critical applications and services, using only one link and ISP provider to access the public network represents a single point of failure. The Link Controller system removes the risk of this single point of failure by enabling enterprises to control and monitor multiple links for their Internet connectivity.

The Link Controller system:

- Guarantees reliable network connections and eliminates downtime by detecting any type of connection outage, and transparently directing traffic away from malfunctioning or unavailable links.
- Distributes traffic to optimize the capacity of each connection by monitoring line throughput so that links do not become over-saturated.
- Increases site performance. The Link Controller system measures and directs users over the best performing link to increase site response times.
- Directs traffic over the least expensive link. Administrators can define the price of links and tiered pricing schemes. The BIG-IP Link Controller system can direct traffic to the least expensive connection, lowering bandwidth costs.
- Controls traffic to match business priorities. Organizations can define traffic policies to direct traffic over specified connections.

Link management

With the Link Controller system, you can manage both inbound and outbound traffic over multiple links. You can distribute traffic based on performance, bandwidth cost, and bandwidth availability. The metrics you can specify are the limits on bandwidth usage, and the pricing structure of

your purchased bandwidth. When you specify the limits and pricing metrics for your links, the Link Controller system then load balances the links based on those metrics.

Link monitoring

You can monitor several aspects of your managed links using the following tools:

- ◆ **Link Statistics**

The Link Statistics screen, in the Configuration utility, displays information about the status, bandwidth usage, bandwidth limits, and bandwidth costs for each of the links managed by the Link Controller system. For more information on link statistics, review Chapter 12, *Viewing Statistics*.

- ◆ **Transparent monitoring**

Transparent monitoring provides the health status of the routers for the managed links.



2

Essential Configuration Tasks

- Introducing essential configuration tasks
- Creating the default gateway pool
- Adding links
- Adding listeners
- Adding pools
- Adding virtual servers
- Adding wide IPs
- Configuring address translation

Introducing essential configuration tasks

When you integrate a Link Controller™ into your network, you must complete a specific set of tasks for the capabilities of the Link Controller to be available to you. These tasks are:

- Create a default gateway pool, which contains the links available for load balancing traffic.
- Add links that the Link Controller manages.
- Add a listener that identifies the traffic that the Link Controller must load balance.
- Add a pool that contains the virtual servers to which the Link Controller load balances inbound traffic.
- Add virtual servers that represent the resources to which the Link Controller load balances inbound traffic.
- Add a wide IP for which the Link Controller load balances inbound traffic.
- Configure address translation settings for inbound and outbound traffic management.

Once you complete these essential configuration tasks, you can customize how the Link Controller system manages inbound and outbound traffic. For example, you can determine if you want to load balance traffic based on cost, bandwidth, or application. You can also define specific geographic regions, called topologies, that allow you to load balance traffic based on the traffic destination.

◆ Note

*The Link Controller uses an auto-discovery feature to manage and maintain links. You can access this feature from the Main tab of the navigation pane by expanding **System**, clicking **General Properties**, and then choosing **General** from the Global Traffic menu. Do not disable this feature; doing so causes the Link Controller to mark all links as down, and to be unable to manage traffic.*

Creating the default gateway pool

One of the most important tasks you want to complete when adding links to the Link Controller system configuration is the creation of a default gateway pool. A **default gateway pool** is a collection of the routers available for handling the network's inbound and outbound traffic. The Link Controller system requires the default gateway pool to load balance traffic across different routers, ensuring that network traffic flows in an efficient and cost-effective manner.

To create a default gateway pool

1. On the Main tab of the navigation pane, expand **Local Traffic** and then click **Pools**.
The Pool List screen opens.
2. Click the **Create** button.
3. In the **Name** box, type the name of the pool.
F5 Networks recommends that you use a name such as **default_gateway_pool**.
Note: The pool name is limited to 63 characters.
4. For the **New Members** setting, add the IP addresses associated with each router, to add the routers to the pool.
5. Configure additional options for the pool as needed.
The system displays additional options when you select **Advanced** from the **Configuration** list.
6. Click **Finished** to save your changes.

Implementing the default gateway pool

After you create a default gateway pool, you must instruct the Link Controller system to use the pool as the default gateway connection between the internal network and the Internet.

To implement the default gateway pool

1. On the Main tab of the navigation pane, expand **Network** and then click **Routes**.
The Route List screen opens.
2. Click the **Add** button.
3. From the **Type** list, select **Default IPv4 Gateway** or **Default IPv6 Gateway**.
4. From the **Resource** list, select **Use Pool**.
5. From the **Pool** list, select the pool name that represents the group of links you want to use as the default gateway pool.
6. Click **Finished** to save your changes.

Adding links

Before you can load balance inbound and outbound traffic on the Link Controller system, you must add at least one link and configure its basic properties. These properties include settings such as the router address of the link, as well as the limit thresholds for inbound and outbound traffic.

To add a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the **Create** button.
3. In the **Name** box, type a name for the link.
Note: The link name is limited to 63 characters.
4. In the **Router Address** box, type the IP address that you want to associate with the link.
5. In the **Uplink Address** box, type the IP address that you want to associate with the Internet Service Provider (ISP) to which outgoing traffic is sent.
6. In the **Service Provider** box, type the name of the Internet Service Provider.
7. Select the health monitors that the Link Controller system uses to determine the availability of the link:
 - a) Click the name of the monitor in the **Available** list.
 - b) Use the Move [<<] button to add the monitor to the **Enabled** list.
F5 Networks recommends that you add at least the **bigip_link** and **snmp_link** monitors.
8. Click the **Create** button to save your changes.

For additional information on adding and managing links, see Chapter 3, *Defining Links*.

Adding listeners

One of the most crucial aspects of integrating the Link Controller system into your network is providing it with the means of identifying the network traffic for which it is responsible. A **listener** is a resource for the Link Controller system that monitors an IP address on which the system intercepts traffic. **Listening** is a process in which a component, known as a listener, passively checks incoming traffic and initiates an action only if a packet matches a set of criteria. Each listener that you define monitors only for DNS packets on port **53**. The Link Controller system then handles only network traffic sent to that IP address.

◆ **Note**

The IP address that you supply for a listener typically is the IP address you assigned to the Link Controller system.

In most situations, a Link Controller system is responsible for traffic that traverses multiple VLANs. Consequently, you can configure a listener to monitor as many or as few VLANs as necessary.

To configure a listener

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. In the **Destination** box, type the IP address on which you want the Link Controller to listen for network traffic.
The Link Controller system handles only network traffic sent to this IP address.
4. From the **VLAN Traffic** list, select a VLAN setting appropriate for this listener.
For additional assistance with this setting, see the online help for this screen.
5. Click the **Finished** button to save the new listener.

Repeat this process for any additional listeners that you want to create.

For additional information on adding and managing listeners, see Chapter 4, *Working with Listeners*.

Adding pools

A **load balancing pool** is a set of devices, such as web servers, that you group together to receive and process traffic. Instead of sending client traffic to the destination IP address specified in the client request, the Link Controller system sends the request to any of the servers that are members of that pool.

To create a load balancing pool

1. On the Main tab of the navigation pane, expand **Local Traffic** and then click **Pools**.
The Pool List screen opens.
2. Click the **Create** button.
3. From the **Configuration** list, select **Advanced**.
The screen displays additional settings.
4. In the **Name** box, type a name for the pool.
Note: The pool name is limited to 63 characters.
5. Specify, retain, or change each of the other settings.
For information on pool settings, refer to the online help for this screen.
6. Click the **Finished** button to save your changes.

For more information on adding and managing pools, see Chapter 5, *Configuring Load Balancing Pools*.

Adding virtual servers

A key requirement for the Link Controller system is that you add the virtual servers to which the Link Controller system load balances inbound and outbound traffic. A virtual server is a specific network resource and, at a minimum, is identified by an IP address and port number.

To add a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic** and then click **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
3. In the **Name** box, type the name of the virtual server.
Note: The virtual server name is limited to 63 characters.
4. For the **Destination** setting:
 - a) Select the appropriate destination type:
 - If the virtual server represents a single resource, select **Host**.
 - If the virtual server represents a group of resources, select **Network**.
 - b) In the **Address** box, type the IP address of the virtual server.
5. In the **Service Port** box, type the port number that the virtual server uses.
Alternatively, you can select a port from the adjacent list. For example, if you select **HTTP** from the list, the corresponding box automatically updates to contain the corresponding port number, **80**.
6. Configure the remaining properties associated with this virtual server as needed.
7. Click the **Finished** button to save the new virtual server configuration.

For additional information on adding and managing virtual servers, see Chapter 6, *Configuring Virtual Servers*.

Adding wide IPs

A **wide IP** is a mapping of a fully-qualified domain name (FQDN) to a set of virtual servers that host the domain's content, such as a web site, an e-commerce site, or a content delivery network (CDN). You establish wide IPs within a Link Controller system to determine how the system manages inbound traffic.

To add a wide IP

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the **Create** button.
3. In the **Name** box, type a name that identifies the wide IP.
4. In the **TTL** box, type the time-to-live value you want to associate with this wide IP.
5. For the **Load Balancing Method** setting, select the appropriate load balancing modes.
You can select a load balancing mode at three different levels: **Preferred**, **Alternate**, and **Fallback**. These modes are described in detail in Chapter 9, *Inbound Load Balancing*.
6. In the **Member List** setting, add the virtual servers that belong to this wide IP.
7. Click the **Create** button to save the new wide IP.

Repeat this process for any additional wide IPs that you want to create.

For more information on adding and managing wide IPs, see Chapter 7, *Defining Wide IPs*.

Configuring address translation

A virtual server configured on a Link Controller system translates the destination IP address of an incoming packet to another destination IP address, for the purpose of load balancing that packet. Normally, the source IP address remains unchanged.

As an option, you can also create a *secure network address translation (SNAT)*. A *SNAT* is an object that maps an original client IP address (that is, a source IP address) to a translation address that you choose. Thus, a SNAT causes the Link Controller system to translate the source IP address of an incoming packet to an address that you specify. The purpose of a SNAT is simple: to ensure that the target server sends its response back through the Link Controller system rather than to the original client IP address directly.

As an alternative to SNAT, the Link Controller system also supports *network address translation (NAT)*. A NAT provides an alias IP address that a node can use as its source IP address when making or receiving connections to clients on the external network. (This distinguishes it from a SNAT, which can initiate but not receive a connection.)

The Link Controller system supports multiple types of SNAT and NAT implementations. The following procedure outlines the most basic SNAT implementation, referred to as a standard implementation, which maps a source IP address directly to another IP address.

To configure secure address translation

1. On the Main tab of the navigation pane, expand **Local Traffic**, and then click **SNATs**.
The SNAT List screen opens.
2. Click the **Create** button.
3. In the **Name** box, type a unique name for the SNAT.
4. From the **Translation** list, select **IP Address**.
5. Type an IP address in the adjacent box.
6. Click the **Finished** button to save your changes.

For additional information on SNAT and NAT configurations, see Chapter 19, *Configuring SNATs and NATs*.



3

Defining Links

- Introducing links
- Adding links
- Adding health monitors
- Setting bandwidth restrictions
- Configuring link weighting and billing properties

Introducing links

Before you can load balance inbound and outbound traffic on the Link Controller™ system, you must add at least one link. A **link** is a network device that connects your internal network with the Internet.

To add a link to the Link Controller system, you must configure certain basic properties. These properties include settings such as the router address of the link, as well as the limit thresholds for inbound and outbound traffic. After you establish these basic properties, you can further customize the link through a variety of advanced configuration options.

One of the most critical aspects of adding links to the Link Controller system is adding each link to a specialized group within the system, called the default gateway pool. The Link Controller system uses this pool to load balance network traffic that is moving between your private network and the Internet at large.

In general, the tasks you accomplish with links are:

- Adding links to the Link Controller system
- Adding health monitors to links to track their availability
- Setting bandwidth restrictions
- Configuring weighting and billing properties

Adding links

Links form the backbone of the Link Controller system configuration, as they determine how traffic enters and leaves your network. The load balancing features of the Link Controller system require at least two links; however, you can add as many links as you need to manage your inbound and outbound traffic.

To add a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the **Create** button.
3. In the **Name** box, type a name for the link.
Note: The link name is limited to 63 characters.
4. In the **Router Address** box, type the IP address that is defined as a member of the default gateway pool. This IP address is also considered the IP address assigned to the router that belongs to the local area network.
5. In the **Uplink Address** box, type the IP address that identifies the router on the wide-area network. The Link Controller system uses this address to obtain SNMP statistics from the routers WAN interface.
6. In the **Service Provider** box, type the name of the Internet Service Provider.
7. For the **Health Monitors** setting that the Link Controller system uses to determine the availability of the link:
 - a) Click the name of the monitor in the **Available** list.
 - b) Use the Move [<<] button to add the monitor to the **Enabled** list.
F5 Networks recommends that you add at least the **bigip_link** and **snmp_link** monitors.
8. Click the **Create** button to save your changes.

Adding health monitors

A **health monitor** is a software utility that tracks the performance and availability of a given link. The Link Controller system uses health monitors to ensure that it is selecting the best link when load balancing network traffic. You can add a health monitor when you first configure a link, or you can add it at a later time. For more detailed information on monitors, including how to configure the properties of existing monitors or create a custom monitor for your configuration, see Chapter 8, *Configuring Monitors*.

To add a health monitor to a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. For the **Health Monitors** setting, select the appropriate monitors from the **Available** list, then use the Move [<<] button to move these monitors to the **Enabled** list.
If you are unsure as to which monitors you should add to the link, F5 Networks recommends that you assign at least the **bigip_link** and **snmp_link** monitors.
4. Click the **Update** button to save your changes.

Monitoring the Link Controller

One of the first monitors you should add to any link is the **bigip_link** monitor. This monitor is responsible for gathering metrics and statistical data that the Link Controller system gathers from other monitors that are verifying the availability of other network resources and services.

For example, the IT department at the fictional company, SiteRequest, is using an FTP and HTTP monitor to track the health of a web server. In addition to these monitors, the IT department also assigns the **bigip_link** monitor, which accumulates the data the FTP and HTTP monitors gather, allowing the IT department to view statistical and health information on the web server through the Configuration utility.

To add a bigip_link monitor to a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.

3. For the **Health Monitors** setting, select the appropriate monitors from the **Available** list, then use the Move [<<] button to move these monitors to the **Enabled** list.
4. Click the **Update** button to save your changes.

Monitoring with SNMP

Another option when determining how to monitor the health of a network resource, such as a web server, is to use the Simple Network Management Protocol, or SNMP. Tracking the health of a resource through SNMP requires that you:

- Enable the given resource to send SNMP traps.
- Assign the **snmp_link** monitor to the given resource in the Configuration utility.

The **snmp_link** monitor is a specialized health monitor that acquires SNMP data that a given resource sends, and makes that information available to you through the Configuration utility.

To add an snmp_link monitor to a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. For the **Health Monitors** setting, select **snmp_link** from the **Available** list, then use the Move [<<] button to move the monitor to the **Enabled** list.
4. Click the **Update** button to save your changes.

Determining availability requirements

By default, the Link Controller system requires that all health monitors assigned to a resource verify that the resource is available for load balancing traffic. However, you can configure the Link Controller system so that only a certain number of monitors must verify that the link is available.

To determine availability requirements of a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. From the **Configuration** list, select **Advanced**.

4. From the **Availability Requirements** list, select the number of monitors that must identify the link as available.
 - Select **All** to require that all monitors identify the link as available.
 - Select **At Least** to specify a minimum number of monitors that must identify the link as available.
5. Click the **Update** button to save your changes.

Setting bandwidth restrictions

Often, the links that you want to manage with the Link Controller system have bandwidth restrictions that you want to maintain to ensure that your network is operating in a cost-effective manner. Through the Link Controller system, you can establish bandwidth restrictions for each link. When the traffic on a link exceeds a specified value, the system marks the link as unavailable, and attempts to use another link.

To set bandwidth restrictions for a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. From the **Configuration** list, select **Advanced**.
4. For the **Traffic Limits** setting, specify the **Inbound**, **Outbound**, and **Total Traffic Limits**. For each option, you have two choices:
 - **None**, which instructs the Link Controller system to ignore this statistic when determining load balancing options.
 - **Up to**, which allows you to specify a value, in megabits per second, that denotes the maximum amount of traffic of that type for the link.
5. Click the **Update** button to save your changes.

Configuring link weighting and billing properties

With the link weighting and billing properties, you can refine the link load balancing and statistics reporting for the links in your configuration. On the Link Weighting screen, you determine how the Link Controller system manages and distributes traffic across the links, using the following settings.

- ◆ **Duplex Billing**

If your ISP uses duplex billing, you can set the **Duplex Billing** option so that the statistics and billing report screens accurately reflect the bandwidth usage for the link. Note that by default, the duplex billing setting is on.

- ◆ **Price Weighting**

If you want the Link Controller system to load balance traffic based on the cost of the bandwidth, then select the **Use Price Weighting** option. You can use this weighting option to avoid the costs associated with exceeding your prepaid bandwidth. You can also use this weighting option to direct traffic over the least expensive link first.

- ◆ **Ratio Weighting**

If you want the Link Controller system to load balance the total traffic to the controller based on a ratio, then select the **Use Ratio Weighting** option. When you have links of varying bandwidth sizes, you can use this weighting option to avoid oversaturating a smaller link with too much traffic.

- ◆ **Important**

You can use either the price weighting option or the ratio weighting option to load balance your link traffic for all of your links. You cannot use both options. Regardless of which weighting option you use, you must use the same weighting option for all links.

Configuring duplex billing for each link

Certain ISPs employ duplex billing when managing the costs of an Internet connection. **Duplex billing** refers to a billing method in which the current traffic rate is considered to be the maximum of either the inbound or outbound traffic. By default, any links that you add in the Link Controller system have the **Duplex Billing** option enabled.

To configure duplex billing for a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. From the **Configuration** list, select **Advanced**.
4. Check **Duplex Billing**, located at the bottom of the screen.
5. Click the **Update** button to save your changes.

Configuring price weighting for each link

You can set the price weighting that you want the Link Controller system to use when load balancing link traffic. The price weighting is based on the billing structure that your ISP uses. The Link Controller system load balances traffic to another link if the lowest cost link reaches a threshold that you have set. This helps you control traffic based on the cost the ISP is charging for the bandwidth.

WARNING

If you configure price weighting for one link on the Link Controller system, you must configure price weighting for all of the remaining links in the configuration. If you do not, the Link Controller system load balances only to the link for which price weighting is defined.

To configure the price weighting for a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. From the **Configuration** list, select **Advanced**.
4. From the **Weighting** list, select **Price (Dynamic Ratio)**.

5. Configure the following cost elements associated with the link:
 - In the **Prepaid Segment** box, type the maximum bandwidth usage you pay for each month. For example, if your ISP charges \$1000 for bandwidth usage in the 0 to 1.54 bps range, type **154**. That means that you pay every month for up to 1.54 bps even if you do not use the link at all.
 - For the **Incremental Segments** setting, type the bandwidth (in bps) and the associated cost of the next pricing tier. For example, if your ISP charges \$2200 for bandwidth usage in the 1.54 to 3 Mbps range, type **300** in the **Up to** box and **2200** in the cost box. Click the **Add** button to add the new cost tier to the configuration. You can add additional cost tiers to the configuration if required.
6. Click the **Update** button to save your changes.

Configuring link ratio weighting for each link

In a configuration in which you have multiple links available to handle inbound and outbound traffic, it is not uncommon for the Link Controller system to determine that one or more links are valid options for load balancing traffic. You can control how the system handles this situation by assigning a link ratio weight for each link. When multiple links are available to handle traffic, the Link Controller system selects a link based on the values of the link ratios of each link.

For example, the fictional company, SiteRequest, has two links available for handling network traffic during business hours. One link is a T1 line, while the other is a DSL line. To ensure their network traffic flows efficiently, the IT team at SiteRequest sets the link ratio value for the T1 link to 2 and the link ratio value for the DSL link to 1. This configuration instructs the Link Controller system to send twice as much traffic through the T1 link as through the DSL link.

If you want to load balance link traffic using link ratio weighting, you need to configure the following settings in addition to specifying the ratio weight on the link.

◆ Inbound load balancing

The inbound load balancing mode, on a wide IP, is **Quality of Service**. All QOS coefficients should be set to **0**, except for the **Link Capacity** coefficient, which should be any number other than **0**.

◆ Outbound load balancing

The outbound load balancing mode, for a router pool (usually the default gateway pool), is **Dynamic Ratio**.

To configure the link ratio weighting for a link

1. On the Main tab of the navigation pane, expand **Link Controller**, and then click **Links**.
The Link List screen opens.
2. Click the name of the link that you want to modify.
3. From the **Configuration** list, select **Advanced**.
4. In the **Link Ratio** box, type a link ratio value.
5. Click the **Update** button to save your changes.



4

Working with Listeners

- Introducing listeners
- Creating a listener for local resolution
- Configuring listeners for traffic forwarding
- Configuring a wildcard listener
- Modifying listeners
- Deleting listeners
- Using listeners with VLANs

Introducing listeners

Before you can fully configure the Link Controller™ system to handle name resolution requests, you must determine how you want to integrate the system into your existing network. Part of this integration includes identifying what network traffic is relevant to the Link Controller system and how you want the system to respond to this traffic. In general, there are two options when handling traffic with the Link Controller system:

- The system receives the traffic, processes it locally, and sends the appropriate DNS response back to the querying server. Link Controller systems with this configuration are considered to be running in ***node mode***.
- The system receives the traffic and forwards it to either another part of the network or another DNS server. Link Controller systems with this configuration are considered to be running in either ***bridge mode*** or ***router mode***, depending on where the system is forwarding network traffic.

To control how the Link Controller system handles network traffic, you configure one or more listeners. A ***listener*** is a specialized resource that is assigned a specific IP address and uses port **53**, the DNS query port. When traffic is sent to that IP address, the listener alerts the system, allowing it to handle the traffic locally or forward the traffic to the appropriate resource.

◆ Tip

If you are familiar with the Local Traffic Manager™, it might be helpful to consider a listener as a specialized type of virtual server that is responsible for handling traffic for the Link Controller system.

You control how the Link Controller system responds to network traffic on a per-listener basis. For example, a single Link Controller system can be the authoritative server for one domain, while forwarding other requests to a separate DNS server. Regardless of how many listeners you configure, the system always manages and responds to requests for any inbound wide IPs that you have configured on it.

Creating a listener for local resolution

Often, when you add a Link Controller system, you want the system to respond to at least a subset of your incoming DNS requests. These requests can be directed at inbound wide IPs that you have configured on the Link Controller system, but you are not limited to wide IPs alone. You can also configure the Link Controller system to respond to DNS requests for other network resources that might not be associated with a wide IP.

When a Link Controller system is operating in node mode, you assign it a listener that corresponds to an IP address on the Link Controller system.

- ◆ If the Link Controller system is operating as a standalone unit, this IP address is the self IP address of the Link Controller system.
- ◆ If the Link Controller is part of a redundant system configuration for high availability purposes, this IP address is the floating IP address that belongs to both systems.

To configure a listener for local resolution

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. In the **Destination** box, type the IP address on which the Link Controller system listens for network traffic.
In this case, add either the self IP address of the system, or, in the case of a redundant system configuration, the floating IP address that corresponds to both systems.
4. From the **VLAN Traffic** list, select a VLAN setting appropriate for this listener.
Typically, if the Link Controller system is handling traffic on this IP address locally, select **All VLANs** for this option
5. Click the **Finished** button to save the new listener.

Configuring listeners for traffic forwarding

Another common configuration you can use with the Link Controller system is to integrate it with your existing DNS servers. In this scenario, the Link Controller system handles any traffic related to the inbound wide IPs you assign to it, while sending other DNS requests to another DNS server on your network. When forwarding traffic in this manner, the system is considered to be operating in bridge or router mode, depending on how the traffic was initially sent to the Link Controller system. In this configuration, you assign a listener to the Link Controller system that corresponds to the IP address of the DNS server to which you want to forward traffic.

Unlike the procedure described in *Creating a listener for local resolution*, on page 4-2, in this procedure you can create more than one listener to forward network traffic. The number of listeners you create depends on your network configuration and the ultimate destination to which you want to send specific DNS requests.

To configure a listener for traffic forwarding

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. In the **Destination** box, type the IP address on which the Link Controller system listens for network traffic.
In this case, the IP address that you add is the IP address of the DNS server that you want to handle the DNS request.
4. From the **VLAN Traffic** list, select a VLAN setting appropriate for this listener.
Typically, if the Link Controller system is handling traffic on this IP address locally, select **All VLANs** for this option.
5. Click the **Finished** button to save the new listener.

Configuring a wildcard listener

In some cases, you might want the Link Controller system to handle any traffic coming into your network, regardless of the destination IP address of the given DNS request. In this configuration, the Link Controller system continues to process and respond to requests for the inbound wide IPs that you configure, but in addition it is responsible for forwarding any other DNS requests to other network resources, such as other DNS servers. To accomplish this type of configuration, you assign a wildcard listener to the system. A *wildcard listener* is the same as a standard listener, except that it contains an asterisk (*) instead of an IP address.

To configure a wildcard listener

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. In the **Destination** box, type an asterisk (*).
4. From the **VLAN Traffic** list, select a VLAN setting appropriate for this listener.
5. Click the **Finished** button to save the new listener.

Modifying listeners

After you create a listener, you can access its settings, changing them as needed. Common instances in which you need to modify a listener include adding an additional VLAN, or modifying the IP address of the listener.

To modify a listener

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners screen opens.
2. Click the name of the listener.
3. Modify the settings for the listener.
4. Click the **Update** button to save your changes to the listener.

Deleting listeners

In the event that you no longer need a specific listener within the Link Controller system, you can delete it.

To delete a listener

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Check the Select box that corresponds to the listener that you want to delete.
3. Click the **Delete** button.
The Delete confirmation screen displays.
4. Click the **Delete** button to delete the listener.

Using listeners with VLANs

One of the features of a BIG-IP® system is that you can create one or more VLANs and assign specific interfaces to the VLANs of your choice. By default, each BIG-IP system includes at least two VLANs, named **internal** and **external**.

When you assign listeners to the Link Controller™ system, you must take into account any VLANs that you have created. For example, a listener that forwards traffic to another DNS server might only be appropriate for a specific VLAN, while a wildcard listener might be applicable to all VLANs. You can configure a listener in one of three ways: to be applicable to all VLANs, enabled only on specific VLANs, or disabled on specific VLANs.

Setting up a listener for all VLANs

If the Link Controller system resides on a network segment that does not use VLANs, or if the IP address you assign as a listener is valid for all VLANs for which the system is responsible, you set the **VLAN Traffic** option to **All VLANs**.

To set up a listener for all available VLANs

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. In the **Destination** box, type the IP address on which the Link Controller system listens for network traffic.
4. From the **VLAN Traffic** list, select **All VLANs**.
5. Click **Finished**.

Enabling a listener for specific VLANs

If the Link Controller system manages traffic for only some of the VLANs available on the network segment, you set the **VLAN Traffic** option to **Enabled on**.

To set up a listener for all available VLANs

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. In the **Destination** box, type the IP address on which the Link Controller system listens for network traffic.
4. From the **VLAN Traffic** list, select **Enabled on**.
5. For the **VLAN List** setting, select the appropriate VLANs from the **Available** list and use the Move buttons (<<) to move them to the **Selected** list.
This listener alerts the Link Controller system only about traffic on the VLANs in the **Selected** list.
6. Click the **Finished** button to save your changes.

Disabling a listener for specific VLANs

In instances where the Link Controller system resides on a network segment with several VLANs, and you want to exclude some VLANs from the listener, you set the **VLAN Traffic** option to **Disabled on**.

To set up a listener for all available VLANs

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Listeners**.
The Listeners List screen opens.
2. Click the **Create** button.
3. From the **VLAN Traffic** list, select **Disabled on**.
4. For the **VLAN List** setting, select the appropriate VLANs from the **Available** list and use the Move buttons (<<) to move them to the **Selected** list.
This listener alerts the Link Controller system about traffic on all VLANs except those listed in the **Selected** list.
5. Click the **Finished** button to save your changes.



5

Configuring Load Balancing Pools

- Introducing load balancing pools
- Creating and modifying load balancing pools
- Configuring pool settings
- Configuring pool member settings
- Managing pools and pool members

Introducing load balancing pools

In a typical client-server scenario, a client request goes to the destination IP address specified in the header of the request. For sites with a large amount of incoming traffic, the destination server can quickly become overloaded as it tries to service a large number of requests. To solve this problem, the BIG-IP® Link Controller™ system distributes client requests to multiple servers instead of to the specified destination IP address only. You configure the BIG-IP system to do this when you create a load balancing pool.

What is a load balancing pool?

A **load balancing pool** is a logical set of devices, such as web servers, that you group together to receive and process traffic. Instead of sending client traffic to the destination IP address specified in the client request, the BIG-IP system sends the request to any of the servers that are members of that pool. This helps to efficiently distribute the load on your server resources.

When you create a pool, you assign pool members to the pool. A **pool member** is a logical object that represents a physical node (server), on the network. You then associate the pool with a virtual server on the BIG-IP system. Once you have assigned a pool to a virtual server, the BIG-IP system directs traffic coming into the virtual server to a member of that pool. An individual pool member can belong to one or multiple pools, depending on how you want to manage your network traffic.

The specific pool member to which the BIG-IP system chooses to send the request is determined by the load balancing method that you have assigned to that pool. A **load balancing method** is an algorithm that the BIG-IP system uses to select a pool member for processing a request. For example, the default load balancing method is **Round Robin**, which causes the BIG-IP system to send each incoming request to the next available member of the pool, thereby distributing requests evenly across the servers in the pool. For a complete list of load balancing methods, see *Specifying the load balancing method*, on page 5-12.

Features of a load balancing pool

You can configure the BIG-IP system to perform a number of different operations for a pool. You can:

- Associate health monitors with pools and pool members
- Enable or disable SNAT connections
- Rebind a connection to a different pool member if the originally-targeted pool member becomes unavailable
- Set the Quality of Service or Type of Service level within a packet

- Specify a load balancing algorithm for a pool
- Assign pool members to priority groups within a pool

Creating and modifying load balancing pools

You use the Configuration utility to create a load balancing pool, or to modify a pool and its members. When you create a pool, the BIG-IP system automatically assigns a group of default settings to that pool and its members. You can retain these default settings or modify them. Also, you can modify the settings at a later time, after you have created the pool.

It is helpful to understand that the BIG-IP system designates some settings as basic and others as advanced. If you decide to modify some of the default settings when you create the pool, be sure to select the **Advanced** option on the screen to view all configurable settings. For more information on basic and advanced settings in general, see Chapter 1, *Introducing the Link Controller*.

Creating and implementing a load balancing pool

Creating and implementing a load balancing pool is a two-task process:

- First, you must create the pool.
- Second, you must associate the pool with a virtual server.

To create a load balancing pool

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the upper-right corner of the screen, click **Create**.
The New Pool screen opens.
3. From the **Configuration** list, select **Advanced**.
4. In the **Name** box, type a name for the pool.
Note: The pool name is limited to 63 characters.
5. Specify, retain, or change each of the other settings.
For information on pool settings, see *Configuring pool settings*, on page 5-6, or refer to the online help for this screen.
6. Click **Finished**.

To implement a load balancing pool

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of the appropriate virtual server.
The settings for the virtual server display.
3. On the menu bar, click **Resources**.
4. From the **Default Pool** list, select the name of your newly-created pool.
5. Click **Update**.

Modifying a load balancing pool

You can modify any settings configured for an existing pool, including the load balancing method. For information on pool settings, see *Configuring pool settings*, on page 5-6, or see the online help. For information on adding members to an existing pool, see *Modifying pool membership*, following.

To modify pool settings

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. Click the name of an existing pool.
The settings for the pool display
3. From the **Configuration** list, select **Advanced**.
4. Modify or retain all settings.
5. Click **Update**.
6. If you want to modify the load balancing method or enable or disable priority group activation, locate the menu bar and click **Members**.
7. Modify or retain the **Load Balancing Method** and **Priority Group Activation** settings.
8. Click **Update**.

Modifying pool membership

For an existing load balancing pool, you can either modify existing pool members or add new members to the pool.

Modifying existing pool members

When modifying settings for members of a pool, you can:

- Enable or disable pool members
- Remove members from the pool
- Modify the values of pool member settings

To modify existing pool members

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the Members column, click the number shown.
3. Under Current Members, modify a pool member:
 - a) If you want to enable or disable a pool member, or remove a member from the pool, click the box to the left of a member address. Then click **Enable**, **Disable**, or **Remove**.
 - b) If you want to modify the settings for a pool member, click an address and retain or modify pool member settings as needed.
For information on pool member settings, see *Configuring pool member settings*, on page 5-16.
4. Click **Update**.

Adding members to an existing load balancing pool

Not only can you specify pool members at the time that you create a pool, you can add pool members later. When adding a pool member to an existing pool (as opposed to specifying a pool member during pool creation), you can configure a number of settings for that pool member. The only settings that you must explicitly specify are the **Address** and **Service Port** settings. All other settings have default values that you can either retain or adjust, depending on your needs.

◆ Note

*If you specify a pool member at the time that you create a pool, you do not see these settings; instead, the BIG-IP system simply assigns default values. However, you can adjust the settings later by modifying the pool member properties. For more information, see **Managing pools and pool members**, on page 5-20.*

To add members to a load balancing pool

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the Members column, click the number shown.
This lists the existing members of the pool.
3. On the right side of the screen, click **Add**.
The New Pool Members screen opens.
4. In the **Address** box, type an IP address.
5. In the **Service Port** box, type a port number or select a service from the list.
6. Retain or configure all other settings. For information on pool member settings, see *Configuring pool member settings*, on page 5-16.
7. Click **Finished**.

Configuring pool settings

You can configure pool settings to tailor pools to your specific needs. For those settings that have default values, you can retain those default settings or modify them. Also, you can modify any settings either when you create the pool, or at any time after you have created it. For information on how to use the Configuration utility to configure these settings, see *Creating and modifying load balancing pools*, on page 5-2.

Table 5.1 lists the settings that you can configure for a pool, followed by a description of each setting.

Pool Setting	Description	Default Value
Name	Specifying a name for a pool is required. The pool name is limited to 63 characters.	No default value
Description	Specify a unique description of this pool.	No default value
Health Monitors	You can associate a health or performance monitor with an entire pool, rather than with individual pool members only. This eases the task of configuring health and performance monitoring for multiple web servers.	No default value
Availability Requirement	You can specify the number of monitors that must report a pool member as being available before that member is defined as being in an up state.	All
Allow SNAT	You can configure a pool so that SNATs are automatically enabled or disabled for any connections using that pool.	Yes
Allow NAT	You can configure a pool so that NATs are automatically enabled or disabled for any connections using that pool.	Yes
Action on Service Down	If this setting is enabled and the target pool member goes down, the BIG-IP system tries to choose another pool member and rebind the client connection to a new server connection. Possible values are None , Reject , Drop , and Reselect .	None
Slow Ramp Time	This option causes the BIG-IP system to send a less-than-normal amount of traffic to a newly-enabled pool member for the specified amount of time.	0
IP ToS to Client	You can configure a pool to set a specific Type of Service (ToS) level within a packet sent to a client, based on the targeted pool.	Pass Through
IP ToS to Server	You can configure a pool to set a specific Type of Service (ToS) level within a packet sent to a server, based on the targeted pool.	Pass Through
Link QoS to Client	You can configure a pool to set a specific Quality of Service (QoS) level within a packet sent to a client, based on the targeted pool.	Pass Through
Link QoS to Server	You can configure a pool to set a specific Quality of Service (QoS) level within a packet sent to a server, based on the targeted pool.	Pass Through

Table 5.1 Settings for a load balancing pool

Pool Setting	Description	Default Value
Reselect Tries	Specifies the number of times the system tries to contact a new pool member after a passive failure. A passive failure consists of a server-connect failure or a failure to receive a data response within a user-specified interval. The default of 0 (zero) indicates no reselects.	0
Enable Request Queueing	Enables TCP request queueing.	No
Request Queue Depth	Specifies the maximum number of connection requests allowed in the queue. The default value of 0 equates to unlimited connection requests, constrained only by available memory.	0
Request Queue Timeout	Specifies the maximum number of milliseconds that a connection request can be queued until capacity becomes available, whereupon the connection request is removed from the queue and reset. The default value of 0 equates to an unlimited time in the queue.	0
Load Balancing Method	You can use the default load balancing method, or you can define another load balancing method, and you can configure priority-based member activation. Different pools can be configured with different load balancing methods.	Round Robin
Priority Group Activation	You can assign pool members to priority groups within the pool.	Disabled
New Members	For each pool that you create, you must specify the servers that are to be members of that pool. Pool members must be specified by their IP addresses. For each pool member, you can also assign a service port, a ratio weight, and a priority group. The pool member name is limited to 63 characters.	No default value

Table 5.1 Settings for a load balancing pool

Before configuring a pool, it is helpful to have a description of certain pool settings that you might want to change.

Specifying a pool name

The most basic setting you can configure for a pool is the pool name. Pool names are case-sensitive and may contain letters, numbers, and underscores (_) only. Reserved keywords are not allowed.

Each pool that you define must have a unique name.

Important

The pool name is limited to 63 characters.

Associating health monitors with a pool

Monitors are a key feature of the BIG-IP system. Monitors help to ensure that a server is in an **up** state and able to receive traffic. When you want to associate a monitor with an entire pool of servers, you do not need to explicitly associate that monitor with each individual server. Instead, you can simply use the pool setting **Health Monitors** to assign the monitor to the pool itself. The BIG-IP system then automatically monitors each member of the pool.

The BIG-IP system contains many different pre-configured monitors that you can associate with pools, depending on the type of traffic you want to monitor. You can also create your own custom monitors and associate them with pools. The only monitor types that are not available for associating with pools are monitors that are specifically designed to monitor nodes and not pools or pool members. These monitor types are:

- ICMP
- TCP Echo
- Real Server
- SNMP DCA
- SNMP DCA Base
- WMI

With the BIG-IP system, you can configure your monitor associations in many useful ways:

- You can associate a monitor with an entire pool instead of an individual server. In this case, the BIG-IP system automatically associates that monitor with all pool members, including those that you add later. Similarly, when you remove a member from a pool, the BIG-IP system no longer monitors that server.
- When a server that is designated as a pool member allows multiple processes to exist on the same IP address and port, you can check the health or status of each process. To do this, you can add the server to multiple pools, and then within each pool, associate a monitor with the that server. The monitor you associate with each server checks the health or performance of the process running on that server.
- When associating a monitor with an entire pool, you can exclude an individual pool member from being associated with that monitor. In this case, you can associate a different monitor for that particular pool member, or you can exclude that pool member from health monitoring altogether. For example, you can associate pool members A, B, and D with the **http** monitor, while you associate pool member C with the **https** monitor.
- You can associate multiple monitors with the same pool. For instance, you can associate both the **http** and **https** monitors with the same pool.

For detailed information on health and performance monitors, see Chapter 8, *Configuring Monitors*.

Specifying the availability requirements

This setting specifies a minimum number of health monitors. Before the BIG-IP system can report the pool member as being in an **up** state, this number of monitors, at a minimum, must report a pool member as being available to receive traffic.

To configure this setting, type a number in the **Availability Requirement** box.

Allowing SNATs and NATs

When configuring a pool, you can specifically disable any secure network address translations (SNATs) or network address translations (NATs) for any connections that use that pool. You do this by configuring the **Allow SNAT** and **Allow NAT** settings. By default, these settings are enabled. You can change this setting on an existing pool by displaying the Properties screen for that pool.

One case in which you might want to configure a pool to disable SNAT or NAT connections is when you want the pool to disable SNAT or NAT connections for a specific service. In this case, you could create a separate pool to handle all connections for that service, and then disable the SNAT or NAT for that pool.

For general information on SNATs and NATs, see Chapter 19, *Configuring SNATs and NATs*.

Specifying action when a service becomes unavailable

The **Action on Service Down** setting specifies the action that you want the BIG-IP system to take when the service on a pool member becomes unavailable. The possible settings are:

- **None:** The BIG-IP system takes no action. This is the default action.
- **Reject:** The BIG-IP system sends an RST or ICMP message.
- **Drop:** The BIG-IP system simply cleans up the connection.

To configure this setting, locate the **Action on Service Down** setting and select a value from the list.

Configuring a slow ramp time

When you take a pool member offline, and then bring it back online, the pool member can become overloaded with connection requests, depending on the load balancing mode for the pool. For example, if you use the Least

Connections load balancing mode, the system sends all new connections to the newly-enabled pool member (because technically, that member has the least amount of connections).

When you configure the **Slow Ramp Time** setting, the system sends less traffic to the newly-enabled pool member. The amount of traffic is based on the ratio of how long the pool member has been available compared to the slow ramp time, in seconds. Once the pool member has been online for a time greater than the slow ramp time, the pool member receives a full proportion of the incoming traffic.

To configure this setting, locate the **Slow Ramp Time** setting and type a number. This number represents the number of seconds that the system waits before sending traffic to a newly-enabled pool member.

Configuring the Type of Service (ToS) level

Another pool setting for a pool is the Type of Service (ToS) level. The **ToS** level is one means by which network equipment can identify and treat traffic differently based on an identifier.

As traffic enters the site, the BIG-IP system can set the ToS level on a packet. Using the **IP ToS to Server** ToS level that you define for the pool to which the packet is sent, the BIG-IP system can apply an iRule and send the traffic to different pools of servers based on that ToS level.

The BIG-IP system can also tag outbound traffic (that is, the return packets based on an **HTTP GET**) based on the **IP ToS to Client** ToS value set in the pool. That value is then inspected by upstream devices and given appropriate priority.

For example, to configure a pool so that a ToS level is set for a packet sent to that pool, you can set both the **IP ToS to Client** level and the **IP ToS to Server** levels to **16**. In this case, the ToS level is set to **16** when sending packets to the client and when sending packets to the server.

◆ Note

*If you change the **ToS** level on a pool for a client or a server, existing connections continue to use the previous setting.*

Configuring the Quality of Service (QoS) level

Another setting for a pool is the Quality of Service (QoS) level. In addition to the ToS level, the **QoS** level is a means by which network equipment can identify and treat traffic differently based on an identifier. Essentially, the QoS level specified in a packet enforces a throughput policy for that packet.

As traffic enters the site, the BIG-IP system can set the QoS level on a packet. Using the **Link QoS to Server** QoS level that you define for the pool to which the packet is sent, the BIG-IP system can apply an iRule that sends the traffic to different pools of servers based on that QoS level.

The BIG-IP system can also tag outbound traffic (that is, the return packets based on an **HTTP GET**) based on the **Link QoS to Client** QoS value set in the pool. That value is then inspected by upstream devices and given appropriate priority.

For example, to configure a pool so that a QoS level is set for a packet sent to that pool, you can set the **Link QoS to Client** level to **3** and the **Link QoS to Server** level to **4**. In this case, the QoS level is set to **3** when sending packets to the client, and set to **4** when sending packets to the server.

Specifying the load balancing method

Load balancing is an integral part of the BIG-IP system. Configuring load balancing on a BIG-IP system means determining your load balancing scenario, that is, which pool member should receive a connection hosted by a particular virtual server. Once you have decided on a load balancing scenario, you can specify the appropriate load balancing method for that scenario.

A **load balancing method** is an algorithm or formula that the BIG-IP system uses to determine the node to which traffic is sent. Individual load balancing methods take into account one or more dynamic factors, such as current connection count. Because each application of the BIG-IP system is unique, and node performance depends on a number of different factors, we recommend that you experiment with different load balancing methods, and select the one that offers the best performance in your particular environment.

Using the default load balancing method

The default load balancing method for the BIG-IP system is Round Robin, which simply passes each new connection request to the next server in line. All other load balancing methods take server capacity and/or status into consideration.

If the equipment that you are load balancing is roughly equal in processing speed and memory, Round Robin mode works well in most configurations. If you want to use the Round Robin method, you can skip the remainder of this section, and begin configuring other pool settings that you want to add to the basic pool configuration.

Selecting a load balancing method

Several different load balancing methods are available for you to choose from. If you are working with servers that differ significantly in processing speed and memory, you may want to switch to one of the ratio or dynamic ratio methods.

◆ Note

On certain hardware platforms, additional load-balancing methods might be available.

◆ Round Robin

This is the default load balancing method. Round Robin mode passes each new connection request to the next server in line, eventually distributing connections evenly across the array of machines being load balanced. Round Robin mode works well in most configurations, especially if the equipment that you are load balancing is roughly equal in processing speed and memory.

◆ **Ratio (member) and Ratio (node)**

The BIG-IP system distributes connections among machines according to ratio weights that you define, where the number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine. These are static load balancing methods, basing distribution on static user-assigned ratio weights that are proportional to the capacity of the servers. Regarding Ratio load balancing:

Load balancing calculations may be localized to each pool (member-based calculation) or they may apply to all pools of which a server is a member (node-based calculation). This distinction is especially important with the Ratio method; with the Ratio (member) method, the actual ratio weight is a member setting in the pool definition, whereas with the Ratio (node) method, the ratio weight is a setting of the node.

The default ratio setting for any node is **1**. If you use the Ratio (as opposed to Ratio (member) load balancing method, you must set a ratio other than **1** for at least one node in the configuration. If you do not change at least one ratio setting, the load balancing method has the same effect as the Round Robin load balancing method.

***Warning:** If you set the load balancing method to Ratio (node), as opposed to Ratio (member), you must define a ratio setting for each node.*

◆ **Dynamic Ratio (member) and Dynamic Ratio (node)**

The Dynamic Ratio method is like the Ratio method except that ratio weights are based on continuous monitoring of the servers and are therefore continually changing.

This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time.

The Dynamic Ratio method is used specifically for load balancing traffic to RealNetworks® RealSystem® Server platforms, Windows® platforms equipped with Windows Management Instrumentation (WMI), or any server equipped with an SNMP agent such as the UC Davis SNMP agent or Windows 2000 Server SNMP agent. To implement Dynamic Ratio load balancing, you must first install and configure the necessary server software for these systems, and then install the appropriate performance monitor. For more information, see Appendix A, *Additional Monitor Considerations*.

◆ **Fastest (node) and Fastest (application)**

The Fastest methods pass a new connection based on the fastest response of all currently active nodes. These methods may be particularly useful in environments where nodes are distributed across different logical networks. Load balancing calculations may be localized to each pool (member-based calculation) or they may apply to all pools of which a server is a member (node-based calculation).

◆ **Least Connections (member) and Least Connections (node)**

The Least Connections methods are relatively simple in that the BIG-IP system passes a new connection to the node that has the least number of current connections. Least Connections methods work best in environments where the servers or other equipment you are load balancing have similar capabilities.

These are dynamic load balancing methods, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time.

Load balancing calculations may be localized to each pool (member-based calculation) or they may apply to all pools of which a server is a member (node-based calculation).

◆ **Observed (member) and Observed (node)**

The Observed methods use a combination of the logic used in the Least Connections and Fastest modes. With the Observed methods, nodes are ranked based on a combination of the number of current connections and the response time. Nodes that have a better balance of fewest connections and fastest response time receive a greater proportion of the connections. The Observed modes also work well in any environment, but may be particularly useful in environments where node performance varies significantly.

These are dynamic load balancing methods, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time.

Load balancing calculations may be localized to each pool (member-based calculation) or they may apply to all pools of which a server is a member (node-based calculation).

◆ **Predictive (member) and Predictive (node)**

The Predictive methods also use the ranking methods used by the Observed methods, where nodes are rated according to a combination of the number of current connections and the response time. However, with the Predictive methods, the BIG-IP system analyzes the trend of the ranking over time, determining whether a node's performance is currently improving or declining. The nodes with better performance rankings that are currently improving, rather than declining, receive a higher proportion of the connections. The Predictive methods work well in any environment.

The Predictive methods are dynamic load balancing methods, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time.

Load balancing calculations may be localized to each pool (member-based calculation) or they may apply to all pools of which a server is a member (node-based calculation).

Specifying priority-based member activation

You can load balance traffic across all members of a pool or across only members that are currently activated according to their priority number. In priority-based member activation, each member in a pool is assigned a priority number that places it in a priority group designated by that number.

With all pool members available (meaning they are enabled, marked **up**, and have not exceeded their connection limit), the BIG-IP system distributes connections to all members in the highest priority group only, that is, the group designated by the highest priority number. The **Priority Group Activation** value determines the minimum number of members that must remain available for traffic to be confined to that group. If the number of available members in the highest priority group drops below the minimum number, the BIG-IP system also distributes traffic to the next higher priority group, and so on. The configuration shown in Figure 5.1 has three priority groups, **3**, **2**, and **1**.

```
pool my_pool {
    lb_mode fastest
    min active members 2
    member 10.12.10.7:80 priority 3
    member 10.12.10.8:80 priority 3
    member 10.12.10.9:80 priority 3
    member 10.12.10.4:80 priority 2
    member 10.12.10.5:80 priority 2
    member 10.12.10.6:80 priority 2
    member 10.12.10.1:80 priority 1
    member 10.12.10.2:80 priority 1
    member 10.12.10.3:80 priority 1
}
```

Figure 5.1 Sample pool configuration for priority load balancing

Connections are first distributed to all pool members with priority **3** (the highest priority group). If fewer than two priority **3** members are available, traffic is directed to the priority **2** members as well. If both the priority **3** group and the priority **2** group have fewer than two members available, traffic is directed to the priority **1** group. The BIG-IP system continuously monitors the higher priority groups, and each time a higher priority group once again has the minimum number of available members, the BIG-IP system again limits traffic to that group.

Specifying pool members

When you configure this setting, you are specifying the servers (that is, pool members) that make up the load balancing pool. To specify a pool member, you use the **Address** and **Service Port** settings to specify the server's IP address, and a service port.

Optional settings are the ratio weight, applicable when you have selected the load balancing method **Ratio (member)**, **Ratio (node)**, or **Dynamic Ratio**, and the priority group activation. For more information on these settings, see *Configuring pool member settings*, on page 5-16.

Configuring pool member settings

When adding members to a pool, you can configure a number of settings for that pool member. You configure most of these settings after you have created the load balancing pool. The only settings that you must specify during pool creation are the **Address** and **Service Port** settings. All other settings have default values that you can either retain or adjust later, depending on your needs.

For information on adding pool members during pool creation, see *Creating and implementing a load balancing pool*, on page 5-2. For information on adding members to an existing pool, see *Adding members to an existing load balancing pool*, on page 5-4.

Table 5.2 shows the settings that you can configure for an existing pool member, followed by a description of each setting.

General property	Description	Default Value
State	Specifies the state of the pool member. You can select one of three options: Enabled , Disabled , or Forced Offline . If you select Forced Offline , active connections continue to use the pool, but new connections do not.	Enabled
Ratio	Specifies the ratio weight that you want to assign to the pool member.	1
Priority Group	Specifies the priority group for the pool member.	1
Connection Limit	Specifies the maximum number of concurrent connections allowed for a pool member.	0

Table 5.2 *Configuration settings for an individual pool member*

General property	Description	Default Value
Health Monitors	Specifies whether you want the pool member to inherit the monitor associated with the pool or to use a different monitor.	Inherit From Pool
Select Monitors	Specifies the monitor or monitors that you want to associate with that pool member. This setting is used only when you set the Health Monitors setting to Member Specific .	No default value
Availability Requirement	Specifies a minimum number of health monitors. Before the BIG-IP system can report the pool member as being in an up state, this number of monitors, at a minimum, must report a pool member as being available to receive traffic.	

Table 5.2 Configuration settings for an individual pool member

To adjust settings for a pool member after you have added it to a pool, see *To view pool member properties*, on page 5-24.

Before adding pool members, it is helpful to have a description of certain pool member settings that you might want to change.

Specifying a ratio weight for a pool member

When using a ratio-based load balancing method for distributing traffic to servers within a pool, you can use the **Ratio** setting to assign a ratio weight to the server. The ratio weight determines the amount of traffic that the server receives.

The ratio-based load balancing methods are: **Ratio (member)**, **Ratio (node)**, and **Dynamic Ratio**. For more information on ratio-based load balancing methods, see *Specifying the load balancing method*, on page 5-12, and Appendix A, *Additional Monitor Considerations*.

Specifying priority-based member activation

The **Priority** setting assigns a priority number to the pool member. Within the pool, traffic is then load balanced according to the priority number assigned to the pool member. Thus, members that are assigned a high priority receive the traffic until the load reaches a certain level, at which time the traffic goes to members assigned to the next lower priority group.

You use the **Priority Group Activation** setting to configure the load level that determines when the BIG-IP system begins directing traffic to members of a lower priority. For more information, see *Specifying priority-based member activation*, on page 5-15.

Specifying a connection limit

With the **Connection Limit** setting, you can specify the maximum number of concurrent connections allowed for a pool member. Note that the default value of **0** (zero) means that there is no limit to the number of concurrent connections that the pool member can receive.

Selecting an explicit monitor association

Once you have associated a monitor with a pool, the BIG-IP system automatically associates that monitor with every pool member, including those members that you add to the pool later. However, in some cases you might want the monitor for a specific pool member to be different from that assigned to the pool. In this case, you must use the **Health Monitors** setting to specify that you want to explicitly associate a specific monitor with the pool member.

You can also configure this setting to prevent the BIG-IP system from associating any monitor with that pool member.

To explicitly associate a monitor with a pool member, locate the **Health Monitors** setting and select **Member Specific**, which causes the **Select Monitors** setting to appear. Then configure the **Select Monitors** setting as described in the following section.

To ensure that the BIG-IP system associates no monitor with the pool member, set the **Health Monitors** setting to **None**.

Creating an explicit monitor association for a pool member

The BIG-IP system contains many different monitors that you can associate with a pool member, depending on the type of traffic you want to monitor. You can also create your own custom monitors and associate them with pool members. The only monitor types that are not available for associating with pool members are monitors that are specifically designed to monitor nodes and not pools or pool members. These monitor types are:

- ICMP
- TCP Echo
- Real Server
- SNMP DCA
- SNMP DCA Base
- WMI

For detailed information on health and performance monitors, see Chapter 8, *Configuring Monitors*.

To associate a monitor with an individual pool member, you simply display the pool member settings and set the **Health Monitors** setting to **Member Specific**. This displays the **Select Monitors** setting. Select the monitor that

you want to associate with the pool member, and using the Move button (<<), move the monitor name to the **Active** box. Clicking **Finished** or **Update** activates the monitor association for that pool member only.

Associating multiple monitors with the same pool member

The BIG-IP system allows you to associate more than one monitor with the same server. Using the Configuration utility, you can:

- ◆ **Associate more than one monitor with a member of a single pool.**
For example, you can create monitors **http1**, **http2**, and **http3**, where each monitor is configured differently, and associate all three monitors with the same pool member. In this case, the pool member is marked as **down** if any of the checks are unsuccessful.
- ◆ **Assign one IP address and service to be a member of multiple pools.**
Then, within each pool, you can associate a different monitor with that pool member. For example, suppose you assign the server **10.10.10:80** to three separate pools: **my_pool1**, **my_pool2**, and **my_pool3**. You can then associate all three custom HTTP monitors to that same server (one monitor per pool). The result is that the BIG-IP system uses the **http1** monitor to check the health of server **10.10.10:80** in pool **my_pool1**, the **http2** monitor to check the health of server **10.10.10:80** in pool **my_pool2**, and the **http3** monitor to check the health of server **10.10.10:80** in pool **my_pool3**.

You can make multiple-monitor associations either at the time you add the pool member to each pool, or by later modifying a pool member's properties.

Specifying an availability requirement

The **Availability Requirement** setting specifies a minimum number of health monitors. Before the BIG-IP system can report the pool member as being in an **up** state, this number of monitors, at a minimum, must report a pool member as being available to receive traffic.

Managing pools and pool members

When generally managing pools and pool members, you typically need to view existing pool or pool member configurations. Occasionally, you might need to perform other management tasks as well. Using the Configuration utility, you can:

- Manage pools
- Manage pool members
- Disable monitor associations for pools and pool members
- View statistics for pools and pool members

An important part of managing pools and pool members is viewing and understanding the status of a pool or pool member at any given time. The Configuration utility indicates status by displaying one of several icons, distinguished by shape and color, for each pool or pool member:

- The shape of the icon indicates the status that the monitor has reported for that pool or pool member. For example, a circular icon indicates that the monitor has reported the pool member as being **up**, whereas a diamond-shaped icon indicates that the monitor has reported the pool member as being **down**.
- The color of the icon indicates the actual status of the node itself. For example, a green shape indicates that the node is **up**, whereas a red shape indicates that the node is **down**. A black shape indicates that user-intervention is required.

To further understand these status icons, see *Understanding pool status*, on page 5-21 and *Understanding pool member status*, on page 5-24.

For information on modifying pool properties, see *Modifying a load balancing pool*, on page 5-3. For information on modifying pool-member properties, see *Modifying existing pool members*, on page 5-4.

Managing pools

There are certain pool-specific tasks that you can perform on the BIG-IP system to maintain existing load-balancing pools. For those pools that you have permission to manage, you can view a list of pools, display the properties of a pool, view the status of a pool, or delete a pool.

Viewing a list of pools

You can view a list of the existing pools that you have permission to view. When you display the list of pools, the Configuration utility displays the following information about each pool:

- Status
- Name
- Number of pool members

Use the following procedure to view a list of pools defined on the BIG-IP system.

To view the list of pools

On the Main tab, expand **Local Traffic**, and click **Pools**.

This opens the Pools screen and displays a list of the pools that you have permission to view.

Viewing pool properties

You can use the Configuration utility to view the general properties of a pool. Note that you can only view properties of those pools that you have permission to view. The pool properties and their descriptions are:

◆ Name

The unique name that you assigned to the pool. An example of a pool name is **my_http_pool**.

◆ Availability

The status of the pool, based on:

- Whether the pool is enabled
- The status of parent nodes
- The status of pool members based on reports from associated health monitors

To view pool properties

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. Click a pool name.
The properties of the pool display.

Understanding pool status

At any time, you can determine the status of a pool. The status of a pool is based solely on the status of its members. Using the Configuration utility, you can find this information by viewing the **Availability** property of the pool. You can also find this information by displaying the list of pools and checking the Status column.

The Configuration utility indicates pool status by displaying one of several icons, distinguished by shape and color. To understand these icons, see Table 5.3. To locate the icons within the Configuration utility, see *To view pool properties*, on page 5-21. For background information on status icons, see *Managing pools and pool members*, on page 5-20.





Status indicator	Explanation
	At least one pool member is available for processing traffic.
	No pool members are currently available but any one of them could become available later, with no user action required. An example of an unavailable pool member becoming available automatically is when the number of concurrent connections to the pool member no longer exceeds the value defined in the pool member's Connection Limit setting.
	All pool members are unavailable and therefore cannot accept traffic. A reason for a pool member being unavailable is that an associated EAV monitor has detected that the pool member is unavailable. When pool status is red, user action is usually required.
	<p>The status of at least one pool member is unknown, and no other pool members are available. Sample reasons for unknown pool-member status are:</p> <p>One or more pool members has no associated monitor.</p> <p>Monitor results are not available yet.</p> <p>The pool member's IP address is misconfigured.</p> <p>The parent node has been disconnected from the network.</p>

Table 5.3 *Explanation of status indicators for pools*

Deleting a pool

To delete an existing pool, use the following procedure. For information on removing individual pool members from a pool, see *Modifying existing pool members*, on page 5-4.

Before deleting a pool, you must first remove the pool as a resource from the virtual server.

To delete a pool

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the left column next to a pool name, check the Select box.
3. Click **Delete**.
The Delete Confirmation screen opens.
4. Click **Delete**.

Managing pool members

There are certain tasks specific to pool members that you can perform on the BIG-IP system to maintain those existing pool members. For those pool members that you have permission to manage, you can view a list of pool members, display the properties of the pool member, view the status of a pool member, enable or disable a pool member, or delete a pool member.

Viewing a list of pool members

You can view a list of the pool members that you have permission to view. When you display the list of pool members, the Configuration utility displays the following information about each member:

- Status
- IP address and service
- Ratio
- Priority group

Use the following procedure to view a list of pool members defined for a pool.

To view a list of pool members

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the Name column, click the name of the relevant pool.
3. On the menu bar, click **Members**.
This lists the members of the pool.

Viewing pool member properties

You can use the Configuration utility to view the general properties of an individual pool member. These properties and their descriptions are:

- ◆ **Address**
The IP address of the associated node.
- ◆ **Service port**
The port number of the relevant service.
- ◆ **Parent node**
The node (IP address) with which the pool member is associated. For example, if the pool member is **10.10.10.22:80**, then the parent node is **10.10.10.22**.
- ◆ **Availability**
The status of the pool member, based on the parent node, the pool, and the monitor with which the pool member is associated.
- ◆ **Health monitors**
The health monitors that are associated with the pool member.
- ◆ **Current connections**
The number of current connections that the pool member has received.
- ◆ **State**
The state of the traffic that you want the pool member to receive.
Possible states are:
 - **Enabled (All Traffic Allowed)**
 - **Disabled (Only persistent or active connections allowed)**
 - **Forced offline (Only active connections allowed)**

To view pool member properties

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the Name column, click the name of the relevant pool.
3. On the menu bar, click **Members**.
The members of the pool display.
4. In the Current Members column, click a pool member IP address and port number.
The properties of the pool member display.

Understanding pool member status

At any time, you can determine the status of a pool member. Using the Configuration utility, you can find this information by viewing the **Availability** property of the pool member. You can also find this information by displaying the list of pool members and viewing the Status column.

The Configuration utility indicates pool member status by displaying one of several icons, distinguished by shape and color. To further understand these status icons, see Table 5.4, following. To view the icons within the Configuration utility, see *To view pool member properties*, on page 5-24. For background information on status icons, see *Managing pools and pool members*, on page 5-20.









Status indicator	Explanation	State property is set to...
	The pool member is set to Enabled , the parent node is up , and a monitor has marked the pool member as up .	Enabled (All Traffic Allowed)
	The pool member is unavailable, but could become available later with no user interaction required. This status occurs when the number of concurrent connections has exceeded the limit defined in the pool member's Connection Limit setting.	Enabled (All Traffic Allowed)
	The pool member is unavailable because either the parent node is down, a monitor has marked the pool member as down , or a user has disabled the pool member.	Enabled (All Traffic Allowed)
	The pool member is set to Disabled , although a monitor has marked the pool member as up .	Disabled (Only persistent or active connections allowed)
	The pool member is set to Disabled and is offline because the parent node is down .	Forced Offline (Only active connections allowed)
	The pool member is set to Disabled and is offline because a user disabled it.	Disabled (Only persistent or active connections allowed)
	The pool member is set to Disabled and is offline because either the parent node is down , or a monitor has marked the pool member as down .	Forced Offline (Only active connections allowed)
	The pool member or node has no monitor associated with it, or no monitor results are available yet.	Enabled (All Traffic Allowed)

Table 5.4 Explanation of status icons for pool members

Enabling or disabling a pool member

You can use the Configuration utility to enable or disable individual pool members. When you enable or disable a pool member, you indirectly set the value of the pool member's **State** property, in the following way:

- ◆ **Enable**
Sets the **State** property of the pool member to **Enabled (All traffic allowed)**.
- ◆ **Disable**
Sets the **State** property of the pool member to **Disabled (Allow persistent and active connections only)**.

Note that the difference between a disabled pool member and a pool member that a monitor reports as **down** is that a disabled pool member continues to process persistent and active connections. Conversely, a pool member reported as **down** processes no connections whatsoever.

The status icons on the pool-member list screen and properties screen indicate whether a pool member is currently enabled or disabled. For more information on pool member status, see *Understanding pool member status*, on page 5-24.

To enable or disable a pool member

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The list of pools that you have permission to view displays.
2. In the Members column, click the number shown.
A list of the members of the pool displays.
3. Locate the IP address and port number of the pool member you want to enable or disable.
4. In the column to the left, check the Select box.
5. At the bottom of the screen, click **Enable** or **Disable**.

Removing a pool member

You have the option of removing a pool member from a pool. When you remove a pool member, you can re-assign it to another pool, and any additional configurations associated with the pool member, such as self IP addresses, remain in the system.

To remove a pool member

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The list of pools that you have permission to view displays.
2. In the Members column, click the number shown.
A list of the members of the pool displays.
3. Locate the IP address and port number of the pool member you want to enable or disable.

4. In the column to the left, check the Select box.
5. At the bottom of the screen, click **Remove**.
The system removes the pool member from the pool.

Deleting a pool member

To delete an existing pool, use the following procedure. For information on removing individual pool members from a pool, see *Modifying existing pool members*, on page 5-4.

Before deleting a pool, you must first remove the pool as a resource from the virtual server.

To delete a pool member

1. On the Main tab, expand **Local Traffic**, and click **Pools**.
The Pool List screen opens.
2. In the Name column, click a pool name.
The properties of the pool display.
3. On the menu bar, click **Members**.
A list of pool members displays.
4. In the left column next to a pool member address, check the Select box.
5. Click **Delete**.
The Delete Confirmation screen displays.
6. Click **Delete**.

Removing monitor associations

You can remove any existing monitor associations for a pool or pool member.

To remove a monitor from a pool, access the properties page for the pool and change the **Health Monitors** setting by moving the monitor name in the **Active** box to the **Available** box.

To remove an explicit monitor association on an individual pool member, access the properties page for the pool member and change the **Health Monitors** setting to either **Inherit from Pool** or **None**. Selecting **None** excludes the pool member from any monitoring that you have configured on that pool.

Viewing pool and pool member statistics

Using the Configuration utility, you can view statistics related to existing pools and pool members.

To view pool and pool member statistics, display the list of existing pools or the list of existing pool members. Then click **Statistics** on the menu bar.

This opens the Statistics screen, which shows statistics for all existing pools and their pool members.

The types of statistics shown are:

- Bits (in and out)
- Packets (in and out)
- Connections (current, maximum, and total)
- Requests



6

Configuring Virtual Servers

- Introducing virtual servers and virtual addresses
- Understanding virtual server types
- Creating a virtual server
- Understanding virtual server and virtual address settings
- Managing virtual servers and virtual addresses

Introducing virtual servers and virtual addresses

A virtual server is one of the most important components of any BIG-IP® local traffic management configuration including the Link Controller system. When you configure a virtual server, you create two BIG-IP system objects: a virtual server and a virtual address.

What is a virtual server?

A **virtual server** is a traffic-management object that is represented by an IP address and a service. A virtual server receives a client request, and instead of sending the request directly to the destination IP address specified in the packet header, sends it to any of several content servers that make up a load balancing pool. Virtual servers increase the availability of resources for processing client requests.

Not only do virtual servers distribute traffic across multiple servers, they also treat varying types of traffic differently, depending on your traffic-management needs. For each type of traffic, such as TCP, UDP, HTTP, and FTP, a virtual server can apply an entire group of settings, to affect the way that the BIG-IP system manages that traffic type.

A virtual server can also enable session persistence for many different traffic types. Through a virtual server, you can set up session persistence for HTTP, SIP, and MSRDP connections, to name a few.

Finally, a virtual server can apply an iRule, which is a user-written script designed to inspect and direct individual connections in specific ways. For example, you can create an iRule that searches the content of a TCP connection for a specific string and, if found, directs the virtual server to send the connection to a specific pool or pool member.

To summarize, a virtual server can do the following:

- Distribute client requests across multiple servers to balance server load
- Apply various behavioral settings to multiple traffic types
- Enable persistence for multiple traffic types
- Direct traffic according to user-written iRules®

You can use virtual servers in any of several distinct ways:

◆ **Directing traffic to a load balancing pool**

A **Standard** virtual server (also known as a **load balancing** virtual server) directs client traffic to a load balancing pool and is the most basic type of virtual server. When you first create the virtual server, you assign an existing default pool to it. From then on, the virtual server automatically directs traffic to that default pool.

◆ **Sharing an IP address with a VLAN node**

You can set up a **Forwarding (Layer 2)** virtual server to share the same IP address as a node in an associated VLAN. To do this, you must perform some additional configuration tasks. These tasks consist of:

creating a VLAN group that includes the VLAN in which the node resides, assigning a self-IP address to the VLAN group, and disabling the virtual server on the relevant VLAN.

◆ **Forwarding traffic to a specific destination IP address**

A **Forwarding (IP)** virtual server is just like other virtual servers, except that a forwarding virtual server has no pool members to load balance. The virtual server simply forwards the packet directly to the destination IP address specified in the client request. When you use a forwarding virtual server to direct a request to its originally-specified destination IP address, the BIG-IP system adds, tracks, and reaps these connections just as with other virtual servers. You can also view statistics for a forwarding virtual servers.

◆ **Increasing the speed of processing Layer 4 traffic**

A **Performance (Layer 4)** virtual server is a virtual server with which you associate a Fast L4 profile. Together, the virtual server and profile increase the speed at which the virtual server processes Layer 4 requests.

When you create a virtual server, you specify the pool or pools that you want to serve as the destination for any traffic coming from that virtual server. You also configure its general properties, some configuration options, and other resources you want to assign to it, such as iRules or session persistence types.

The section *Understanding virtual server types*, on page 6-3, describes the types of virtual servers you can create, as well as their general properties, configuration options, and resources.

What is a virtual address?

A **virtual address** is the IP address with which you associate a virtual server. For example, if a virtual server's IP address and service are **10.10.10.2:80**, then the IP address **10.10.10.2** is a virtual address.

You can create a many-to-one relationship between virtual servers and virtual addresses. For example, you can create the three virtual servers **10.10.10.2:80**, **10.10.10.2:443**, and **10.10.10.2:161** for the same virtual address, **10.10.10.2**.

You can enable and disable a virtual address. When you disable a virtual address, none of the virtual servers associated with that address can receive incoming network traffic.

You create a virtual address indirectly when you create a virtual server. When this happens, the BIG-IP system internally associates the virtual address with a MAC address. This in turn causes the BIG-IP system to respond to Address Resolution Protocol (ARP) requests for the virtual address, and to send gratuitous ARP requests and responses with respect to the virtual address. As an option, you can disable ARP activity for virtual addresses, in the rare case that ARP activity affects system performance. This most likely occurs only when you have a large number of virtual addresses defined on the system.

Understanding virtual server types

There are two distinct types of virtual servers that you can create: host virtual servers and network virtual servers.

Host virtual servers

A *host virtual server* represents a specific site, such as an Internet web site or an FTP site, and it load balances traffic targeted to content servers that are members of a pool.

The IP address that you assign to a host virtual server should match the IP address that Domain Name System (DNS) associates with the site's domain name. When the BIG-IP system receives a connection request for that site, the BIG-IP system recognizes that the client's destination IP address matches the IP address of the virtual server, and subsequently forwards the client request to one of the content servers that the virtual server load balances.

Network virtual servers

A *network virtual server* is a virtual server whose IP address has no bits set in the host portion of the IP address (that is, the host portion of its IP address is **0**). There are two kinds of network virtual servers: those that direct client traffic based on a range of destination IP addresses, and those that direct client traffic based on specific destination IP addresses that the BIG-IP system does not recognize.

Directing traffic for a range of destination IP addresses

With an IP address whose host bit is set to **0**, a virtual server can direct client connections that are destined for an entire range of IP addresses, rather than for a single destination IP address (as is the case for a host virtual server). Thus, when any client connection targets a destination IP address that is in the network specified by the virtual server IP address, the BIG-IP system can direct that connection to one or more pools associated with the network virtual server.

For example, the virtual server can direct client traffic that is destined for any of the nodes on the **192.168.1.0** network to a specific load balancing pool such as **ingress-firewalls**. Or, a virtual server could direct a web connection destined to any address within the subnet **192.168.1.0/24**, to the pool **default_webservers**.

Directing traffic for transparent devices (wildcard virtual servers)

Besides directing client connections that are destined for a specific network or subnet, a network virtual server can also direct client connections that have a specific destination IP address that the virtual server does not recognize, such as a transparent device. This type of network virtual server is known as a wildcard virtual server.

Wildcard virtual servers are a special type of network virtual server designed to manage network traffic that is targeted to transparent network devices. Examples of transparent devices are firewalls, routers, proxy servers, and cache servers. A wildcard virtual server manages network traffic that has a destination IP address unknown to the BIG-IP system.

Handling unrecognized client IP addresses

A host-type of virtual server typically manages traffic for a specific site. When the BIG-IP system receives a connection request for that site, the BIG-IP system recognizes that the client's destination IP address matches the IP address of the virtual server, and it subsequently forwards the client to one of the content servers that the virtual server load balances.

However, when load balancing transparent nodes, the BIG-IP system might not recognize a client's destination IP address. The client might be connecting to an IP address on the other side of the firewall, router, or proxy server. In this situation, the BIG-IP system cannot match the client's destination IP address to a virtual server IP address.

Wildcard network virtual servers solve this problem by not translating the incoming IP address at the virtual server level on the BIG-IP system. For example, when the BIG-IP system does not find a specific virtual server match for a client's destination IP address, the BIG-IP system matches the client's destination IP address to a wildcard virtual server, designated by an IP address of **0.0.0.0**. The BIG-IP system then forwards the client's packet to one of the firewalls or routers that the wildcard virtual server load balances, which in turn forwards the client's packet to the actual destination IP address.

Understanding default and port-specific wildcard servers

There are two kinds of wildcard virtual servers that you can create:

- ◆ **Default wildcard virtual servers**

A **default wildcard virtual server** is a wildcard virtual server that uses port **0** and handles traffic for all services. A wildcard virtual server is enabled for all VLANs by default. However, you can specifically disable any VLANs that you do not want the default wildcard virtual server to support. Disabling VLANs for the default wildcard virtual server is done by creating a VLAN disabled list. Note that a VLAN disabled list applies to default wildcard virtual servers only. You cannot create a VLAN disabled list for a wildcard virtual server that is associated with one VLAN only. For the procedure to create a default wildcard server, see *Creating a wildcard virtual server*, on page 6-7.

◆ **Port-specific wildcard virtual servers**

A *port-specific wildcard virtual server* handles traffic only for a particular service, and you define it using a service name or a port number. You can use port-specific wildcard virtual servers for tracking statistics for a particular type of network traffic, or for routing outgoing traffic, such as HTTP traffic, directly to a cache server rather than a firewall or router. For the procedure to create a port-specific wildcard virtual server, see *To create a port-specific wildcard virtual server*, on page 6-8.

If you use both a default wildcard virtual server and port-specific wildcard virtual servers, any traffic that does not match either a standard virtual server or one of the port-specific wildcard virtual servers is handled by the default wildcard virtual server.

We recommend that when you define transparent nodes that need to handle more than one type of service, such as a firewall or a router, you specify an actual port for the node and turn off port translation for the virtual server.

Creating multiple wildcard servers

You can define multiple wildcard virtual servers that run simultaneously. Each wildcard virtual server must be assigned to an individual VLAN, and therefore can handle packets for that VLAN only.

In some configurations, you need to set up a wildcard virtual server on one side of the BIG-IP system to load balance connections across transparent devices. You can create another wildcard virtual server on the other side of the BIG-IP system to forward packets to virtual servers receiving connections from the transparent devices and forwarding them to their destination.

Creating a virtual server

Using the Configuration utility, you can either create a virtual server or modify the settings of an existing virtual server. The following sections contain the procedures for creating and modifying virtual servers. To understand individual virtual server properties and settings, see *Understanding virtual server and virtual address settings*, on page 6-9. For information on viewing existing virtual server configurations, see *Managing virtual servers and virtual addresses*, on page 6-15.

When you create a virtual server, you can create a virtual server that uses many default values for its settings. This makes the task of creating a virtual server fast and easy, because it vastly reduces the number of settings you must explicitly configure.

When creating a virtual server, you can specify the virtual server to be either a host virtual server or a network virtual server. (For more information on host and network virtual servers, see *Host virtual servers*, on page 6-3 and *Network virtual servers*, on page 6-3.) In either case, you need only configure a few settings: a unique name for the virtual server, a destination address, and a service port. If the virtual server is a network type of virtual server, you must also configure the destination type, and a netmask.

◆ Important

The virtual server name is limited to 63 characters.

To create a virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. On the upper right portion of the screen, click the **Create** button. The New Virtual Server screen opens.
3. Configure the required settings.
For more information, see Table 6.1, on page 6-9.
4. Retain or change the values of all other settings.
For information on these settings, see *Understanding virtual server and virtual address settings*, on page 6-9.
5. Click **Finished**.

◆ Note

In a redundant system configuration, you cannot create a virtual server for unit 2 unless you have first created a virtual server for unit 1.

◆ Note

If a virtual server is to have the same IP address as a node in an associated VLAN, you must perform some additional configuration tasks. These tasks consist of: creating a VLAN group that includes the VLAN in which the node resides, assigning self-IP addresses to the VLAN group, and disabling the virtual server on the relevant VLAN.

Creating a wildcard virtual server

A wildcard virtual server is a special type of network virtual server. Creating a wildcard virtual server requires three tasks:

- First, you must create a pool that contains the addresses of the transparent devices.
- Next, you must create the wildcard virtual server (default or port-specific).
- Finally, you must ensure that port translation is disabled for each virtual server. Port translation is disabled by default.

The following procedures describe how to perform these tasks using the Configuration utility. For more information on wildcard virtual servers, see *Directing traffic for transparent devices (wildcard virtual servers)*, on page 6-4.

To create a pool of transparent devices

To create a pool of transparent devices, display the Pools screen and click the **Create** button. For more information, see Chapter 5, *Configuring Load Balancing Pools*.

To create a default wildcard virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. On the upper right portion of the screen, click the **Create** button. The New Virtual Server screen opens.
3. Configure all required settings.
Remember to type the IP address **0.0.0.0** in the **Destination Address** box, and if you selected a network type of virtual server, to type the netmask **0.0.0.0** in the **Mask** box.
4. Click **Finished**.

To create a port-specific wildcard virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. On the upper-right portion of the screen, click **Create**. The New Virtual Server screen opens.
3. For the **Destination** setting, in the **Address** box, type the wildcard IP address **0.0.0.0**.
4. For the **Service Port** setting, type a port number, or select a service name from the list. Note that port **0** defines a wildcard virtual server that handles all types of services. If you specify a port number, you create a port-specific wildcard virtual server. The wildcard virtual server handles traffic only for the port specified.
5. From the **Default Pool** list in the Resources section, select the pool of transparent devices that you want to apply to the virtual server.
6. Click **Finished**.

To turn off port translation for a wildcard virtual server

After you define the wildcard virtual server with a wildcard port, you should verify that port translation is disabled for the virtual server.

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. In the Name column, click the virtual server for which you want to turn off port translation. The Virtual Servers screen opens.
3. From the **Configuration** list, select **Advanced**.
4. Clear the **Port Translation** check box.
5. Click **Update**.

Understanding virtual server and virtual address settings

A virtual server and its virtual server address have a number of properties and settings that you can configure to affect the way that a virtual server manages traffic. You can also assign certain resources to a virtual server, such as a load balancing pool and a persistence profile. Together, these properties, settings, and resources represent the definition of a virtual server or its address, and most have default values. When you create a virtual server, you can either retain the default values or adjust them to suit your needs.

The following sections list and describe all properties, configuration settings, and resources that define virtual servers and virtual addresses.

For information on how to create virtual server, see *Creating a virtual server*, on page 6-6.

Configuring virtual server settings

In the Configuration utility, virtual server settings are grouped into three categories: General properties, configuration settings (basic and advanced), and resources (basic and advanced). The following sections describe the settings that these three categories contain.

General properties

When you create a virtual server, you define some general properties. Table 6.1 lists and describes these general properties.

Property	Description	Default Value
Name	This property is required. The virtual server name is limited to 63 characters.	No default value
Destination Type	The type of virtual server you want to create and its IP address. If the type you select is network , then this property also includes the mask for the IP address. For more information on virtual server types, see <i>Understanding virtual server types</i> , on page 6-3. This property is required.	Host
Destination Address	The IP address of the virtual server.	No default value

Table 6.1 General properties of a virtual server

Property	Description	Default Value
Destination Mask	The netmask for a network virtual server. This property applies to a network virtual server only, and is required. The netmask clarifies whether the host bit is an actual zero or a wildcard representation.	No default value
Service Port	A service name or port number for which you want to direct traffic. This property is required.	No default value
State	The state of the virtual server, that is, Enabled or Disabled . As an option, you can enable or disable a virtual server for a specific VLAN. Note that when you disable a virtual server, the virtual server no longer accepts new connection requests. However, it allows current connections to finish processing before going to a down state. Note: If no VLAN is specified, then the Enabled or Disabled setting applies to all VLANs.	Enabled

Table 6.1 General properties of a virtual server

Configuration settings

When creating a virtual server, you can configure a number of settings. Table 6.2 lists and describes these virtual server configuration settings. Because all of these settings have default values, you are not required to change these settings.

Setting	Description	Default Value
Type	The type of virtual server configuration. Choices are: Standard , IP Forwarding (IP) , Forwarding (Layer 2) , Performance (HTTP) , Performance (Layer 4) , and Reject . For more information, see <i>Introducing virtual servers and virtual addresses</i> , on page 6-1. Note that if set to Reject , this setting causes the BIG-IP system to reject any traffic destined for the virtual server IP address.	Standard
Protocol	The network protocol name for which you want the virtual server to direct traffic. Sample protocol names are TCP and UDP . One benefit of this feature is that you can load balance virtual private network (VPN) client connections across several VPNs, eliminating the possibility of a single point of failure. A typical use of this feature is for load balancing multiple VPN gateways in an IPSEC VPN sandwich, using non-translating virtual servers. An important point to note is that although address translation of such protocols can be optionally activated, some protocols, such as IPSEC in AH mode, rely on the IP headers remaining unchanged. In such cases, you should use non-translating network virtual servers. Note that this setting is disabled when creating a Performance (HTTP) type of virtual server.	TCP
Protocol Profile (Client)	A setting that designates the selected profile as a client-side profile. Applies to TCP and UDP connections only. When creating a Performance (HTTP) type of virtual server, this value is set to fasthttp , and you cannot change it. Similarly, when creating a Performance (Layer 4) type of virtual server, this value is set to fastl4 , and you cannot change it. For more information, see Chapter 16, <i>Managing Protocol Profiles</i> .	TCP

Table 6.2 Configuration settings for a virtual server

Setting	Description	Default Value
Protocol Profile (Server)	A setting that designates the selected profile as a server-side profile. Applies to TCP and UDP connections only. Note that this setting does not appear when creating a Performance (HTTP) or Performance (Layer 4) type of virtual server. For more information, see Chapter 16, <i>Managing Protocol Profiles</i> .	(Use Client Profile)
HTTP Profile	The name of an existing HTTP profile for managing HTTP traffic. Note that this setting does not appear when creating a Performance (HTTP) or Performance (Layer 4) type of virtual server. For more information, see Chapter 14, <i>Managing HTTP and FTP Traffic</i> .	None
FTP Profile	The name of an existing FTP profile for managing FTP traffic. Note that this setting does not appear when creating a Performance (HTTP) or Performance (Layer 4) type of virtual server. For more information, see Chapter 14, <i>Managing HTTP and FTP Traffic</i> .	None
XML Profile	The name of an existing XML profile to be used for XML content-based routing. XML profiles define the matching content to look for in the XML document.	None
SSL Profile (Client)	The name of an existing SSL profile for managing client-side SSL traffic.	None
SSL Profile (Server)	The name of an existing SSL profile for managing server-side SSL traffic.	None
Diameter Profile	The name of an existing Diameter profile.	None
Statistics Profile	The name of a statistics profile. For more information, see Chapter 17, <i>Using the Statistics Profile</i> .	stats
VLAN and Tunnel Traffic	The names of VLANs and tunnels for which the virtual server is enabled or disabled.	ALL VLANs
Rate Class	The name of an existing rate class, used for enforcing a throughput policy for incoming network traffic. For more information, see Chapter 21, <i>Configuring Rate Shaping</i> .	None
Connection Limit	The maximum number of concurrent connections allowed for the virtual server. Setting this to 0 turns off connection limits.	0
Address Translation	A setting to enable or disable address translation on a BIG-IP system. This option is useful when the BIG-IP system is load balancing devices that have the same IP address. This is typical with the nPath routing configuration where duplicate IP addresses are configured on the loopback device of several servers.	Enabled (checked)
Port Translation	A setting to enable or disable port translation on a BIG-IP system. Turning off port translation for a virtual server is useful if you want to use the virtual server to load balance connections to any service.	Enabled (checked)
SNAT Pool	The name of an existing SNAT pool, used for implementing selective and intelligent SNATs. For more information, see Chapter 19, <i>Configuring SNATs and NATs</i> .	None

Table 6.2 Configuration settings for a virtual server

Setting	Description	Default Value
Source Port	A setting that specifies whether the system preserves the source port of the connection.	Preserve
Last Hop Pool	A setting that directs reply traffic to the last hop router using a last hop pool. This overrides the auto_lasthop setting. In cases where you have more than one router sending connections to the BIG-IP system, connections are automatically sent back through the same router from which they were received when the auto_lasthop global variable is enabled, as it is by default. If you want to exclude one or more routers from auto-lasthop , or if the global auto_lasthop is disabled for any reason (for example, you may not want it for an SSL gateway), you can use a last hop pool instead. (If auto_lasthop is enabled, the last hop pool takes precedence over it.) Before configuring a last hop pool, you must first create a pool containing the router inside addresses.	None
NAT64	A setting that specifies whether the system translates IPv6 to IPv4 addresses	Disabled

Table 6.2 Configuration settings for a virtual server

Resources

In addition to assigning various traffic profiles to a virtual server, you can also assign a pool, an iRule, and two persistence profiles. The pool, iRule, and persistence profiles that you assign to a virtual server are known as **resources**.

If you have created a virtual server that is a load balancing type of virtual server, one of the resources you must assign to the virtual server is a default load balancing pool. A **default pool** is the pool to which the BIG-IP system sends traffic if no iRule exists specifying a different pool. Note that if you plan on using an iRule to direct traffic to a pool, you must assign the iRule as a resource to the virtual server.

Table 6.3 lists and describes the specific resources that you can assign to a load balancing virtual server.

Resource	Description	Default Value
iRules	A list of existing iRules that you want the virtual server to use when load balancing its connections. Note that for all iRules that you select, you must configure a corresponding profile on the virtual server. For example, if you are specifying an iRule that includes HTTP commands, you must configure a default or custom HTTP profile on the virtual server. If the iRule you want to implement does not appear in the iRules list, the iRule does not exist and you must first create it. If the iRules setting does not appear on the New Virtual Server screen, check your licensing. For more information on iRules, see Chapter 18, <i>Writing iRules</i> .	No default value
Default Pool	The pool name that you would like the virtual server to use as the default pool. A load balancing virtual server sends traffic to this pool automatically, unless an iRule directs the server to send the traffic to another pool instead. For more information, see Chapter 5, <i>Configuring Load Balancing Pools</i> .	No default value
Default Persistence Profile	The type of persistence that you want the BIG-IP system to use. For more information, see Chapter 15, <i>Enabling Session Persistence</i> .	None
Fallback Persistence Profile	The type of persistence that the BIG-IP system should use if it cannot use the specified default persistence. For more information, see Chapter 15, <i>Enabling Session Persistence</i> .	None

Table 6.3 Resources assigned to a load balancing virtual server

Configuring virtual address settings

The Configuration utility displays virtual address properties and settings. Table 6.4 lists and describes the general properties and configuration settings of a virtual address.

Property	Description	Default Value
Address	The IP address of the virtual server, not including the service.	No default value
Unit ID	The ID of the redundant-pair unit to which this address should apply.	1
Availability	The availability of the virtual address with respect to service checking.	No default value
State	The state of the virtual address, that is, enabled or disabled .	Enabled
Advertise Route	The virtual-server conditions for which the BIG-IP system should advertise this virtual address to an advanced routing module. This setting only applies when the Route Advertisement setting is enabled (checked). Possible values are: When any virtual server is available When all virtual server(s) are available Always	When any virtual server is available

Table 6.4 General properties and configuration settings of a virtual address

Property	Description	Default Value
Connection Limit	The number of concurrent connections that the BIG-IP system allows on this virtual address.	0
ARP	A setting that enables or disables ARP requests for the virtual address. When disabled, the BIG-IP system ignores ARP requests that other routers send for this virtual address.	Enabled (checked)
Route Advertisement	A setting that inserts a route to this virtual address into the kernel routing table so that an advanced routing module can redistribute that route to other routers on the network.	Enabled (checked)

Table 6.4 *General properties and configuration settings of a virtual address*

Managing virtual servers and virtual addresses

When generally managing virtual servers and virtual addresses, you typically need to view existing virtual server or virtual address configurations. Occasionally, too, you might need to delete a virtual server.

When working with virtual servers that you have created, you can:

- View or modify a virtual server configuration.
- View or modify a virtual address configuration.
- View virtual server and virtual address status.
- Enable or disable a virtual server or virtual address.
- Delete a virtual server or virtual address.

Viewing or modifying a virtual server configuration

Occasionally, you might want to determine whether you need to adjust virtual server settings, or create new virtual servers. When you view a virtual server configuration, you can:

- View a list of virtual servers.
- View or modify virtual server properties and settings.
- View virtual server resources.
- View virtual server statistics.

Viewing a list of virtual servers

You can view a list of existing virtual servers that you have permission to view. When you display the list of virtual servers, the Configuration utility displays the following information about each virtual server:

- Status
- Virtual server name
- Destination (virtual address)
- Service port
- Type of virtual server
- Resources (associated pool, HTTP Class profile, iRules, and persistence profiles)

To view a list of virtual servers

On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. This opens the Virtual Servers screen and displays a list of virtual servers.

Viewing or modifying virtual server properties

You can view virtual server properties, such as the profile types that are assigned to the virtual server. Note that you can only view the properties of those virtual servers that you have permission to view.

To view or modify virtual server properties

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. In the Name column, click a virtual server name. The properties of the virtual server display.
3. If you want to modify a virtual server property:
 - a) Locate the property on the screen and change the value.
 - b) Click **Update**.

Viewing or modifying virtual server resources

You can view the default pool, default persistence profile, and fallback persistence profile that are assigned as resources to the virtual server. You can also view any iRules associated with the virtual server. The following procedure shows how to view these resources.

To view or modify virtual server resources

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. Click a virtual server name. The properties of the virtual server display.
3. On the menu bar, click **Resources**.
4. In the Load Balancing section, retain or modify any virtual server resources.
5. Click **Update**.
6. If you want to modify the assignment of an iRule or HTTP Class profile, click the appropriate **Manage** button.
7. Use the **Move** button (<< or >>) to enable or disable an existing iRule or HTTP Class profile.
8. Click **Finished**.

Viewing virtual server statistics

Using the Configuration utility, you can view statistics for any existing virtual servers.

To view statistics for a virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. In the Name column, click the name of a virtual server.
3. On the menu bar, click **Statistics**.
The statistics for the virtual server display.

Viewing or modifying a virtual address configuration

Occasionally, you might want to view or modify virtual address settings. In working with virtual address configurations, you can:

- View a list of virtual addresses.
- View or modify virtual address properties.
- View virtual address statistics.

Viewing a list of virtual addresses

You can view a list of existing virtual addresses that you have created, and adjust any of their settings. When you display the list of virtual addresses, the Configuration utility also displays the state of that address (**enabled** or **disabled**).

To view a list of virtual addresses

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. On the menu bar, click **Virtual Address List**.
A list of virtual addresses displays.

Viewing or modifying virtual address properties

The following procedure shows how to view virtual address properties.

To view or modify virtual address properties

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. On the menu bar, click **Virtual Address List**.
A list of virtual addresses displays.

3. In the Address column, click a virtual address.
The properties of the virtual address display.
4. If you want to modify a virtual address property:
 - a) Locate the property on the screen and change the value.
 - b) Click **Update**.

Viewing virtual address statistics

Using the Configuration utility, you can view statistics for any existing virtual addresses.

To view statistics for a virtual address

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. On the menu bar, click **Virtual Address List**.
The list of existing virtual addresses displays.
3. From the Statistics menu, choose Virtual Server or Virtual Address.
The screen displays statistics of the type you selected.

Understanding virtual server and virtual address status

At any time, you can determine the status of a virtual server or virtual address, using the Configuration utility. You can find this information by displaying the list of virtual servers or virtual addresses and viewing the Status column, or by viewing the **Availability** property of the object.

The Configuration utility indicates status by displaying one of several icons, distinguished by shape and color:

- The shape of the icon indicates the status that the monitor has reported for that node.
- The color of the icon indicates the actual status of the node.

To understand these icons with respect to status, see Table 6.5. To display the icons within the Configuration utility, see *To view or modify virtual server properties*, on page 6-16 and *To view or modify virtual address properties*, on page 6-17.






Status indicator	Explanation
	The virtual server or virtual address is enabled and able to receive traffic.
	<p>The virtual server or virtual address is enabled but is currently unavailable. However, the virtual server or virtual address might become available later, with no user action required.</p> <p>An example of a virtual server or virtual address showing this status is when the object's connection limit has been exceeded. When the number of connections falls below the configured limit, the virtual server or virtual address becomes available again.</p>
	The virtual server or virtual address is enabled but offline because an associated object has marked the virtual server or virtual address as unavailable. To change the status so that the virtual server or virtual address can receive traffic, you must actively enable the virtual server or virtual address.
	The virtual server or virtual address is operational but set to Disabled .
	The status of the virtual server or virtual address is unknown.

Table 6.5 Status icons for virtual servers and virtual addresses

Enabling or disabling a virtual server or virtual address

Using the Configuration utility, you can enable or disable a virtual server or virtual address at any time. When you disable a virtual server or virtual address, the BIG-IP system no longer processes any traffic targeted for that virtual server or virtual address.

You can enable or disable a virtual server or virtual address by first displaying the corresponding list screen from within the Configuration utility.

To enable or disable a virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. In the Name column, click the name of the virtual server you want to enable or disable.
3. Locate the **State** setting.
This setting indicates whether the virtual server is currently enabled or disabled.
4. Return to the list screen.
5. In the Select column, click the box corresponding to the virtual server name.
6. Click **Enable** or **Disable**.

To enable or disable a virtual address

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. On the menu bar, click **Virtual Address List**.
The list of existing virtual addresses displays.
3. In the Address column, locate the virtual address you want to enable or disable.
The State column indicates whether the virtual address is currently enabled or disabled.
4. In the Select column, click the box corresponding to the virtual address.
5. Click **Enable** or **Disable**.

Deleting a virtual server or virtual address

You can permanently delete a virtual server or a virtual address from a configuration. When you delete a virtual server, you automatically delete the corresponding virtual address, if no other virtual servers are associated with that virtual address.

To delete a virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Server List screen opens.
2. Check the Select box to the left of the virtual server that you want to delete.

3. Click **Delete**.
The Delete Confirmation screen opens.
4. Click **Delete**.
The virtual server is deleted.

To delete a virtual address

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. On the menu bar, click **Virtual Address List**.
3. Check the Select box to the left of the virtual address that you want to delete.
4. Click **Delete**.
The Delete Confirmation screen opens
5. Click **Delete**.
The virtual address is deleted.



7

Defining Wide IPs

- Introducing wide IPs
- Configuring wide IPs
- Using wildcard characters in wide IP names
- Modifying a wide IP

Introducing wide IPs

After you configure the virtual servers that the Link Controller™ system manages, you need to group the configured virtual servers into wide IPs. A **wide IP** is a mapping of a fully-qualified domain name (FQDN) to a set of virtual servers that host the domain's content, such as a web site, an e-commerce site, or a CDN.

Before defining the first wide IP, you should gather your configuration information for the Link Controller system so you can easily see which virtual servers have the content you want to map to a domain name.

Configuring wide IPs

After you determine which virtual servers you should place in which wide IP, you are ready to add the first wide IP to the configuration.

To define a wide IP using the Configuration utility

1. On the Main tab of the navigation pane, click **Link Controller** and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the **Create** button.
The New Wide IP screen opens.
3. In the **Name** box, type the identifying name for this wide IP.
4. For the **Load Balancing Method** setting, select the appropriate preferred, alternate, and fallback load balancing methods.
For more information on load balancing, see Chapter 9, *Inbound Load Balancing*.
5. For the **Member List** setting, select the virtual servers that you want to add to the wide IP.
6. Click the **Create** button to save your changes.

Repeat this process for each wide IP you want to add.

Using wildcard characters in wide IP names

The Link Controller system supports wildcard characters in both wide IP names and wide IP aliases. You can use the wildcard characters to simplify your maintenance tasks if you have a large quantity of wide IP names and/or wide IP aliases. The wildcard characters you can use are: the question mark (?), and the asterisk (*). The guidelines for using the wildcard characters are as follows:

◆ **The question mark (?)**

- You can use the question mark to replace a single character, with the exception of dots (.).
- You can use more than one question mark in a wide IP name or alias.
- You can use both the question mark and the asterisk in the same wide IP name or alias.

◆ **The asterisk (*)**

- You can use the asterisk to replace multiple consecutive characters, with the exception of dots (.).
- You can use more than one asterisk in a wide IP name or alias.
- You can use both the question mark and the asterisk in the same wide IP name or alias.

The following examples are all valid uses of the wildcard characters for the wide IP name, **www.mydomain.net**.

- **???.mydomain.net**
- **www.??domain.net**
- **www.my*.net**
- **www.??*.net**
- **www.my*.***
- **???.my*.***
- ***.*.net**
- **www.*.???**

Modifying a wide IP

Once you have defined the basic settings for a wide IP, you can modify the wide IP settings, the load balancing properties, and the virtual server members at any time.

Modifying the basic wide IP settings

When you first add a wide IP to the configuration, you configure the basic settings. You may later decide to modify those settings.

To modify an existing wide IP using the Configuration utility

1. On the Main tab of the navigation pane, click **Link Controller** and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the name of the wide IP that you want to modify.
The properties screen opens.
3. Modify the wide IP settings as needed.
4. Click **Update** to save your changes.



8

Configuring Monitors

- Introducing monitors
- Creating a custom monitor
- Configuring monitor settings
- Special configuration considerations
- Associating monitors with resources
- Managing monitors

Introducing monitors

An important feature of the Link Controller™ system is a set of load-balancing tools called monitors. **Monitors** verify connections on pools and virtual servers. A monitor can be either a health monitor or a performance monitor, designed to check the status of a pool, or virtual server on an ongoing basis, at a set interval. If a pool or virtual server being checked does not respond within a specified timeout period, or the status of a pool or virtual server indicates that performance is degraded, and then the Link Controller system can redirect the traffic to another resource.

Some monitors are included as part of the Link Controller system, while other monitors are user-created. Monitors that the Link Controller system provides are called **pre-configured monitors**. User-created monitors are called **custom monitors**. For more information on pre-configured and custom monitors, see *Understanding pre-configured and custom monitors*, on page 8-4.

Before configuring and using monitors, it is helpful to understand some basic concepts regarding monitor types, monitor settings, and monitor implementation.

◆ Monitor types

Every monitor, whether pre-configured or custom, is a certain type of monitor. Each type of monitor checks the status of a particular protocol, service, or application. For example, one type of monitor is HTTP. An HTTP type of monitor allows you to monitor the availability of the HTTP service on a pool, pool member, or virtual server. A WMI type of monitor allows you to monitor the performance of a pool, or virtual server that is running the Windows Management Instrumentation (WMI) software. An ICMP type of monitor simply determines whether the status of a resource is **up** or **down**. For more information on monitor types, see *Summary of monitor types*, on page 8-2 and *Configuring monitor settings*, on page 8-7.

◆ Monitor settings

Every monitor consists of settings with values. The settings and their values differ depending on the type of monitor. In some cases, the Link Controller system assigns default values. For example, the following are the default values for the ICMP-type monitor:

- Interval: 30 seconds
- Timeout: 120 seconds
- Transparent: No

These settings specify that an ICMP type of monitor is configured to check the status of an IP address every 30 seconds, and to time out every 120 seconds. For more information on monitor settings, see *Summary of status types*, on page 8-3 and *Configuring monitor settings*, on page 8-7.

◆ Monitor implementation

The task of implementing a monitor varies depending on whether you are using a pre-configured monitor or creating a custom monitor. If you want to implement a pre-configured monitor, you need only associate the

monitor with a pool or virtual server. If you want to implement a custom monitor, you must first create the custom monitor, and then associate it with a pool or virtual server. For more information on implementing a monitor, see *Understanding pre-configured and custom monitors*, on page 8-4 and *Creating a custom monitor*, on page 8-6.

Summary of monitor types

The Link Controller system includes many different types of monitors, each designed to perform a specific type of monitoring. The monitors fall into three categories: simple, extended content verification (ECV), and extended application verification (EAV). **Simple monitors** check the health of a resource by sending a packet using the specified protocol, and waiting for a response from the resource. If the monitor receives a response, then the health check is successful and the resource is considered **up**. **ECV monitors** check the health of a resource by sending a query for content using the specified protocol, and waiting to receive the content from the resource. If the monitor receives the correct content, then the health check is successful and the resource is considered **up**. **EAV monitors** check the health of a resource by exercising the specified application. If the monitor receives the correct response, then the health check is successful and the resource is considered **up**.

Table 8.1 describes the types of monitors that you can apply to your load balancing resources. Note that some monitors are available for global traffic-related resources, such as wide IPs, while others are available for local traffic-related resources, such as virtual servers.

Monitor Category	Traffic Type	Monitor Type	Description
Simple	Local Traffic	ICMP	Checks the status of a resource, using Internet Control Message Protocol (ICMP).
Simple	Local Traffic	Gateway ICMP	Checks nodes in a pool that implements gateway failsafe for high availability.
Simple	Local Traffic	TCP Echo	Checks the status of a resource, using Transmission Control Protocol (TCP).
ECV	Local Traffic	HTTP	Verifies the Hypertext Transfer Protocol (HTTP) service by attempting to receive specific content from a web page.
ECV	Local Traffic	HTTPS	Verifies the Hypertext Transfer Protocol Secure (HTTPS) service by attempting to receive specific content from a web page protected by Secure Socket Layer (SSL) security.
ECV	Local Traffic	TCP	Verifies the Transmission Control Protocol (TCP) service by attempting to receive specific content from a resource.

Table 8.1 Monitor types available on a Link Controller system

Monitor Category	Traffic Type	Monitor Type	Description
EAV	Global Traffic	BIG-IP	Acquires data captured through monitors managed by a BIG-IP® Local Traffic Manager™.
EAV	Global Traffic	BIG-IP Link	Acquires data captured through monitors managed by a BIG-IP® Link Controller™ system.
EAV	Local Traffic	FTP	Verifies the File Transfer Protocol (FTP) service by attempting to download a specific file to the <code>/var/tmp</code> directory on the system. Once downloaded successfully, the file is not saved.
EAV	Global Traffic	SNMP Link	Acquires data captured through SNMP traps.

Table 8.1 Monitor types available on a Link Controller system

Summary of status types

When a monitor attempts to verify the availability of a resource, it returns a status of that resource to the Link Controller system. This status displays as a status code in the Configuration utility. A status code is a specific color that denotes the health of a given resource. Using these status codes, you can quickly determine the health of any monitored resource.

The types of status codes available for a resource are:

- **Blue.** A blue status code indicates that the resource has not been checked. This status often displays when you first add a resource into the Configuration utility.
- **Green.** A green status code indicates that the resource is available and operational. The Link Controller system uses this resource to manage traffic as appropriate.
- **Red.** A red status code indicates that the resource did not respond as expected to the monitor. The Link Controller system considers resources with a red status code as down and does not use that resource when managing network traffic.
- **Yellow.** A yellow status code indicates that the resource is operational, but has exceeded one of its established bandwidth thresholds. The Link Controller system only uses a link that has a yellow status code if no other link is available (for example, all other links have a status code of red or blue).
- **Gray.** A gray status code indicates that the resource is unavailable for configuration through this Link Controller system. For example, if a wide IP was removed from the Configuration utility, but its corresponding configuration settings had not been deleted, the link would have a gray status code.

Understanding pre-configured and custom monitors

When you want to monitor the health or performance of pool members or virtual servers, you can either use a pre-configured monitor, or create and configure a custom monitor.

Using pre-configured monitors

For a subset of monitor types, the Link Controller system includes a set of pre-configured monitors. A **pre-configured monitor** is an existing monitor that the system provides for you, with its settings already configured. You cannot modify pre-configured monitor settings, as they are intended to be used as is. The purpose of a pre-configured monitor is to eliminate the need for you to explicitly create one. You use a pre-configured monitor when the values of the settings meet your needs, as is.

The names of the pre-configured monitors that the Link Controller system includes are:

- **big-ip** (global traffic-related resources only)
- **big-ip link** (global traffic-related resources only)
- **gateway icmp** (local traffic-related resources only)
- **http** (local traffic-related resources only)
- **https** (local traffic-related resources only)
- **icmp** (local traffic-related resources only)
- **snmp link** (global traffic-related resources only)
- **tcp** (local traffic-related resources only)
- **tcp_echo** (local traffic-related resources only)

An example of a pre-configured monitor is the **icmp** monitor. If the default values of this monitor meet your needs, you simply assign the **icmp** pre-configured monitor directly to a pool or virtual server. In this case, you do not need to use the Monitors screens, unless you simply want to view the default settings of the pre-configured monitor.

If you do not want to use the values configured in a pre-configured monitor, you can create a custom monitor.

Using custom monitors

A **custom monitor** is a monitor that you create based on one of the allowed monitor types. You create a custom monitor when the values defined in a pre-configured monitor do not meet your needs, or no pre-configured monitor exists for the type of monitor you are creating. (For information on monitor types, see *Summary of monitor types*, on page 8-2.)

Importing settings from a pre-configured monitor

If a pre-configured monitor exists that corresponds to the type of custom monitor you are creating, you can import the settings and values of that pre-configured monitor into the custom monitor. You are then free to change those setting values to suit your needs. For example, if you create a custom monitor called **my_icmp**, the monitor can inherit the settings and values of the pre-configured monitor **icmp**. This ability to import existing setting values is useful when you want to retain some setting values for your new monitor but modify others.

The following list shows an example of a custom ICMP-type monitor called **demo_icmp**, which is based on the pre-configured monitor **icmp**. Note that the Interval value has been changed to **60**. The other settings retain the values defined in the pre-configured monitor.

- Name: demo_icmp
- Type: ICMP
- Interval: 60
- Timeout: 180
- Transparent: No

Importing settings from a custom monitor

You can import settings from another custom monitor instead of from a pre-configured monitor. This is useful when you would rather use the setting values defined in another custom monitor, or when no pre-configured monitor exists for the type of monitor you are creating. For example, if you create a custom monitor called **my_oracle_server2**, you can import settings from an existing Oracle-type monitor such as **my_oracle_server1**. In this case, because the Link Controller system does not provide a pre-configured Oracle-type monitor, a custom monitor is the only kind of monitor from which you can import setting values.

Selecting a monitor is straightforward. Like **icmp**, each of the monitors has a **Type** setting based on the type of service it checks, for example, **http**, **https**, **ftp**, **pop3**, and takes that type as its name. (Exceptions are port-specific monitors, like the **external** monitor, which calls a user-supplied program.)

For procedures on selecting and configuring a monitor, see *Creating a custom monitor*, on page 8-6.

Importing settings from a monitor template

If no pre-configured or custom monitor exists that corresponds to the type of monitor you are creating, the Link Controller system imports settings from a monitor template. A **monitor template** is an abstraction that exists within the system for each monitor type, and contains a group of settings and default values. A monitor template merely serves as a tool for the Link Controller system to use for importing settings to a custom monitor when no monitor of that type already exists.

Creating a custom monitor

When you create a custom monitor, you use the Configuration utility to: give the monitor a unique name, specify a monitor type, and, if a monitor of that type already exists, import settings and their values from the existing monitor. You can then change the values of any imported settings.

You must base each custom monitor on a monitor type. When you create a monitor, the Configuration utility displays a list of monitor types. To specify a monitor type, select the one that corresponds to the service you want to check. For example, if you want to create a monitor that checks the health of the HTTP service on a pool, you choose **HTTP** as the monitor type.

If you want to check more than one service on a pool or virtual server (for example **HTTP** and **HTTPS**), you can associate more than one monitor on that pool or virtual server. For more information, see Chapter 9, *Inbound Load Balancing*.

Checking services is not the only reason for implementing a monitor. If you want to verify only that the destination IP address is alive, or that the path to it through a transparent virtual server is alive, use one of the simple monitors, **icmp** or **tcp_echo**. Or, if you want to verify TCP only, use the monitor **tcp**.

◆ Note

*Before creating a custom monitor, you must decide on a monitor type. For information on monitor types, see **Configuring monitor settings**, on page 8-7.*

To create a custom monitor

1. On the Main tab of the navigation pane, expand **Link Controller** or **Local Traffic** and then click **Monitors**.
The Monitor List screen opens.
2. Click the **Create** button.
The create monitor screen opens.
3. In the **Name** box, type a name for the monitor.
4. From the **Type** list, select the type of monitor that you want to create.
If a monitor of that type already exists, **Import Settings** displays.
5. From the **Configuration** list, select **Advanced**. This allows you to modify additional default settings.
6. Configure all settings shown.
7. Click the **Finished** button to save your changes.

Configuring monitor settings

Before you can create a custom monitor, you must select a monitor type. Monitors types fall into three categories:

- ◆ **Simple monitors**
These are health monitors that monitor the status of a resource.
- ◆ **Extended Content Verification (ECV) monitors**
These are health monitors that verify service status by retrieving specific content from pool members or virtual servers.
- ◆ **External Application Verification (EAV) monitors**
These are health or performance monitors that verify service status by executing remote applications, using an external service-checker program.

Simple monitors

Simple monitors are those that check the status of a resource. The simple monitor types are:

- ICMP
- Gateway ICMP
- TCP Echo
- BIG-IP Link
- SNMP Link

The Link Controller system provides a set of pre-configured simple monitors: **icmp**, **gateway_icmp**, **tcp_echo**, and **tcp_half_open**. You can either use these pre-configured monitors as is, or create custom monitors of these types.

The following sections describe each type of simple monitor and show the pre-configured monitor for each type. Note that each pre-configured monitor consists of settings and their values.

ICMP

Using an ICMP type of monitor, you can use Internet Control Message Protocol (ICMP) to make a simple resource check. The check is successful if the monitor receives a response to an ICMP_ECHO datagram. The following list shows the settings and their values for the pre-configured monitor **icmp**:

- Name: ICMP
- Type: ICMP
- Interval: 5 seconds
- Timeout: 16 seconds

- Transparent: No
- Alias Address: * All Addresses

The **Transparent** mode is an option for ICMP-type monitors. When you set this mode to **Yes**, the monitor pings the resource with which the monitor is associated. For more information about **Transparent** mode, refer to *Using transparent and reverse modes*, on page 8-14.

Gateway ICMP

A Gateway ICMP type of monitor has a special purpose. You use this monitor for a pool that implements gateway failsafe for high availability.

A Gateway ICMP monitor functions the same way as an ICMP monitor, except that you can apply a Gateway ICMP monitor to a pool. (Remember that you can apply an ICMP monitor to a resource only and not a pool member.) The following list shows the settings and their values for the pre-configured **gateway_icmp** monitor.

- Name: Gateway ICMP
- Type: Gateway ICMP
- Interval: 5 seconds
- Timeout: 16 seconds
- Transparent: No
- Alias Address: * All Addresses
- Alias Service Port: * All Ports

TCP Echo

With a TCP Echo type of monitor, you can verify Transmission Control Protocol (TCP) connections. The check is successful if the Link Controller system receives a response to a TCP Echo message. The TCP Echo type also supports **Transparent** mode. In this mode, the resource with which the monitor is associated is pinged through to the destination resource. (For more information about **Transparent** mode, see *Using transparent and reverse modes*, on page 8-14.)

To use a TCP Echo monitor type, you must ensure that TCP Echo is enabled on the resources being monitored. The following list shows the settings for the pre-configured monitor **tcp_echo**:

- Name: TCP Echo
- Type: TCP Echo
- Interval 5 seconds
- Timeout 16 seconds
- Alias Address: * All Addresses

BIG-IP Link

If you employ the Link Controller system in a network that already contains a Link Controller system, you must assign a BIG-IP Link monitor to the system. In fact, this monitor is automatically assigned to the Link Controller system if you do not do so manually.

The BIG-IP Link monitor gathers metrics and statics information that the Link Controller system acquires through the monitoring of its own resources.

The following list shows the settings and default values of a BIG-IP Link-type monitor:

- Name: my_bigip_link
- Type: BIG-IP Link
- Interval: 10 seconds
- Timeout: 30 seconds
- Probe Interval: 1 second
- Probe Timeout: 1 second
- Probe Attempts: 1
- Minimum Required Successful Attempts: 1
- Alias Address: * All Addresses
- Alias Service Port: * All Ports

SNMP Link

You use an SNMP Link type of monitor to check the performance of links that are running an SNMP agent.

The Link Controller system provides a pre-configured SNMP monitor named **snmp_link**. The following list shows the settings and values of the **snmp_link** pre-configured monitor:

- Name: snmp_link
- Type: SNMP Link
- Interval: 10 seconds
- Timeout: 30 seconds
- Probe Interval: 1 second
- Probe Timeout: 1 second
- Probe Attempts: 1
- Minimum Required Successful Attempts: 1
- Alias Addresses: * All Addresses
- Alias Service Port: * All Ports

Performance monitors are generally used with dynamic ratio load balancing. For more information on performance monitors and dynamic ratio load balancing, see Chapter 9, *Inbound Load Balancing*.

◆ **Note**

Unlike health monitors, performance monitors do not report on the status of pool, pool member, or virtual server.

Pre-configured monitors are not user-modifiable. Thus, if you want to change the values for the SNMP Link monitor settings, you must create an SNMP Link-type custom monitor.

Extended Content Verification (ECV) monitors

ECV monitors use **Send String** and **Receive String** settings in an attempt to retrieve explicit content from resources. The Link Controller system provides the pre-configured monitors **tcp**, **http**, and **https** for these ECV monitor types:

- TCP
- HTTP
- HTTPS

You can either use the pre-configured ECV monitors as is, or create custom monitors from these monitor types.

The following sections describe each type of ECV monitor and show the pre-configured monitor for each type. Note that each pre-configured monitor consists of settings and their values.

TCP

A TCP type of monitor attempts to receive specific content sent over TCP. The check is successful when the content matches the **Receive String** value. A TCP type of monitor takes a **Send String** value and a **Receive String** value. If the **Send String** value is blank and a connection can be made, the service is considered **up**. A blank **Receive String** value matches any response. Both **Transparent** and **Reverse** modes are options. For more information about **Transparent** and **Reverse** modes, see *Using transparent and reverse modes*, on page 8-14.

The following list shows the settings for the pre-configured monitor **tcp**:

- Name: tcp
- Type: TCP
- Interval: 5 seconds
- Timeout: 16 seconds
- Send String: "" (empty)
- Receive String: "" (empty)

- Reverse: No
- Transparent: No
- Alias Address: * All Addresses
- Alias Service Port: * All Ports

HTTP

You can use an HTTP type of monitor to check the status of Hypertext Transfer Protocol (HTTP) traffic. Like a TCP monitor, an HTTP monitor attempts to receive specific content from a web page, and unlike a TCP monitor, may send a user name and password. The check is successful when the content matches the **Receive String** value. An HTTP monitor uses a send string, a receive string, a user name, a password, and optional **Reverse** and **Transparent** modes. (If there is no password security, you must use blank strings [""] for the **Username** and **Password** settings.)

For more information on **transparent** and **reverse** modes, see *Using transparent and reverse modes*, on page 8-14.

The following list shows the settings of the pre-configured monitor **http**:

- Name: http
- Type: HTTP
- Interval: 5 seconds
- Timeout: 16 seconds
- Send String: Get /
- Receive String: "" (empty)
- User Name: "" (empty)
- Password: "" (empty)
- Reverse: No
- Transparent: No
- Alias Address: * All Addresses
- Alias Service Port: * All Ports

HTTPS

You use an HTTPS type of monitor to check the status of Hypertext Transfer Protocol Secure (HTTPS) traffic. An HTTPS type of monitor attempts to receive specific content from a web page protected by SSL security. The check is successful when the content matches the **Receive String** value.

HTTPS-type monitors use a send string, a receive string, a user name, a password, and an optional **Reverse** setting. (If there is no password security, you must use blank strings [""] for the **Username** and **Password** settings.) For more information on the **Reverse** setting, see *Using transparent and reverse modes*, on page 8-14.

HTTP-type monitors also include the settings **Cipher List**, **Compatibility**, and **Client Certificate**. If you do not specify a cipher list, the monitor uses the default cipher list **DEFAULT:+SHA:+3DES:+kEDH**. When you set the **Compatibility** setting to **Enabled**, this sets the SSL options to **ALL**. You use the **Client Certificate** setting to specify a certificate file that the monitor then presents to the server.

The following list shows the settings of the pre-configured monitor **https**:

- Name: https
- Type: HTTPS
- Interval: 5 seconds
- Timeout: 16 seconds
- Send String: Get /
- Receive String: "" (empty)
- Cipher List: "" (empty)
- User Name: "" (empty)
- Password: "" (empty)
- Compatibility: Enabled
- Client Certificate: "" (empty)
- Reverse: No
- Alias Address: * All Addresses
- Alias Service Port: * All Ports

The **Reverse** mode is an option for monitors that import settings from the **https** monitor. For more information on **Reverse** mode, see *Using transparent and reverse modes*, on page 8-14.

FTP monitor

Using an FTP type of monitor, you can monitor File Transfer Protocol (FTP) traffic. A monitor of this type attempts to download a specified file to the **/var/tmp** directory, and if the file is retrieved, the check is successful.

◆ Note

Once the file has been successfully downloaded, the Link Controller system does not save it.

An FTP monitor specifies a user name, a password, and a full path to the file to be downloaded.

The following list shows the settings and default values of an FTP-type monitor:

- Name: my_ftp
- Type: FTP
- Interval: 10 seconds

- Timeout: 31 seconds
- User Name: "" (empty)
- Password: "" (empty)
- Path/Filename: "" (empty)
- Mode: Passive
- Alias Addresses: * All Addresses
- Alias Service Port: * All Ports
- Debug: No

Special configuration considerations

Every pre-configured or custom monitor has settings with some default values assigned. The following sections contain information that is useful when changing these default values.

Setting destinations

By default, the value for the **Alias Address** setting for most monitors is set to the wildcard * **Addresses**, and the **Alias Service Port** setting is set to the wildcard * **Ports** (exceptions to this rule are the WMI and Real Server monitors). This value causes the monitor instance created for a pool or virtual server to take that resource's address or address and port as its destination. You can, however, replace either or both wildcard symbols with an explicit destination value, by creating a custom monitor. An explicit value for the **Alias Address** and/or **Alias Service Port** setting is used to force the instance destination to a specific address and/or port which may not be that of the pool or virtual server.

The ECV monitors **http**, **https**, and **tcp** have the settings **Send String** and **Receive String** for the send string and receive expression, respectively.

The most common **Send String** value is **GET /**, which retrieves a default HTML page for a web site. To retrieve a specific page from a web site, you can enter a **Send String** value that is a fully qualified path name:

```
"GET /www/support/customer_info_form.html"
```

The **Receive String** expression is the text string the monitor looks for in the returned resource. The most common **Receive String** expressions contain a text string that is included in a particular HTML page on your site. The text string can be regular text, HTML tags, or image names.

The sample **Receive** expression below searches for a standard HTML tag:

```
"<HEAD>"
```

You can also use the default null **Receive String** value [""]. In this case, any content retrieved is considered a match. If both the **Send String** and **Receive String** are left empty, only a simple connection check is performed.

For HTTP and FTP monitors, you can use the special settings **get** or **hurl** in place of **Send String** and **Receive String** statements. For FTP monitors specifically, the **GET** setting specifies the full path to the file to retrieve.

Using transparent and reverse modes

The normal and default behavior for a monitor is to ping the destination pool or virtual server by an unspecified route, and to mark the resource **up** if the test is successful. However, with certain monitor types, you can specify a route through which the monitor pings the destination server. You configure this by specifying the **Transparent** or **Reverse** setting within a custom monitor.

◆ **Transparent setting**

Sometimes it is necessary to ping the destination through a transparent pool or virtual server. When you create a custom monitor and set the **Transparent** setting to **Yes**, the Link Controller system forces the monitor to ping through the pool or virtual server with which it is associated (usually a firewall) to the pool or virtual server. (In other words, if there are two firewalls in a load balancing pool, the destination pool or virtual server is always pinged through the pool or virtual server specified and not through the pool or virtual server selected by the load balancing method.) In this way, the transparent pool or virtual server is tested: if there is no response, the transparent pool or virtual server is marked as **down**.

Common examples are checking a router, or checking a mail or FTP server through a firewall. For example, you might want to check the router address **10.10.10.53:80** through a transparent firewall **10.10.10.101:80**. To do this, you create a monitor called **http_trans** in which you specify **10.10.10.53:80** as the monitor destination address, and set the **Transparent** setting to **Yes**. Then you associate the monitor **http_trans** with the transparent firewall (**10.10.10.101:80**).

This causes the monitor to check the address **10.10.10.53:80** through **10.10.10.101:80**. (In other words, the Link Controller system routes the check of **10.10.10.53:80** through **10.10.10.101:80**.) If the correct response is not received from **10.10.10.53:80**, then **10.10.10.101:80** is marked **down**. For more information on associating monitors with virtual servers, see *Associating monitors with resources*, on page 8-15.

◆ **Reverse setting**

With the **Reverse** setting set to **Yes**, the monitor marks the pool or virtual server **down** when the test is successful. For example, if the content on your web site home page is dynamic and changes frequently, you may want to set up a reverse ECV service check that looks for the string **"Error"**. A match for this string means that the web server was **down**.

Figure 8.2 shows the monitors that contain the **Transparent** setting, the **Reverse** setting, or both.

Monitor Type	Setting	
TCP	Transparent	Reverse
HTTP	Transparent	Reverse
HTTPS		Reverse
TCP Echo	Transparent	
ICMP	Transparent	

Table 8.2 Monitors that contain the Transparent or Reverse settings

Associating monitors with resources

Once you have created a monitor and configured its settings, the final task is to associate the monitor with the resources to be monitored. The resources can be either a pool or virtual server, depending on the monitor type.

Some monitor types are designed for association with virtual servers only, and not pools. Other monitor types are intended for association with pools only, and not virtual servers. Therefore, when you use the Configuration utility to associate a monitor with a pool or virtual server, the utility displays only those pre-configured monitors that are designed for association with that server. For example, you cannot associate the monitor **icmp** with a pool, since the **icmp** monitor is designed to check the status of a virtual server itself and not any service running on that resource.

When you associate a monitor with a server, the Link Controller system automatically creates an **instance** of that monitor for that server. A monitor association thus creates an instance of a monitor for each server that you specify. Therefore, you can have multiple instances of the same monitor running on your servers.

The Configuration utility allows you to disable an instance of a monitor that is running on a server. This allows you to suspend health or performance checking, without having to actually remove the monitor association. When you are ready to begin monitoring that server again, you simply re-enable that instance of the monitor.

Types of monitor associations

You can assign monitors to different types of objects in the Link Controller system. Different monitors can apply to the same resource, depending on the number of functions that resource has within the network. For example, a monitor assigned to a pool member is checking the same resource as a monitor assigned to a virtual server. In the first instance, the monitor validates whether the resource meets the availability requirements for the pool. In the second instance, the monitor validates whether the resource is available in general.

To illustrate this example, consider a virtual server that contains the home page of a company web site. This virtual server is a member of a pool of servers, all of which can serve the home page on request. To monitor that the virtual server can display the home page, you assign an HTTP monitor to that pool member that checks if it can retrieve the home page. To monitor that the virtual server itself is operational, you assign a TCP monitor it, which checks to see if the virtual server responds to TCP requests. These two monitors, while assigned to the same resource, perform different functions in assessing the availability of different aspects of the network.

The types of monitor associations are:

- ◆ **Monitor-to-pool association**

This type of association links a monitor with an entire load balancing pool. In this case, the monitor checks all members of the pool. For example, you can create an instance of the monitor **http** for the pool **my_pool**, thus ensuring that all members of that pool are checked.

- ◆ **Monitor-to-pool member association**

This type of association links a monitor with a pool member within a given pool. For example, you can create an instance of the monitor **FTP** for specific pools within the pool **my_pool**, ensuring that only specific pool members are verified as available through the FTP monitor.

- ◆ **Monitor-to-virtual server association**

This type of association links a monitor with a specific virtual server. In this case, the monitor checks only the virtual server itself, and not any services running on that virtual server. For example, you can create an instance of the monitor **icmp** for virtual server **10.10.10.10**. In this case, the monitor checks the specific virtual server only, and not any services running on that virtual server.

Managing monitors

When managing existing monitors, you can display or delete them, or you can enable and disable an instance of a monitor. Note that prior to deleting a monitor, you must remove all existing monitor associations.

To display a monitor

1. On the Main tab of the navigation pane, expand **Link Controller** or **Local Traffic** and then click **Monitors**.
The Monitor List screen opens.
2. Click a monitor name.
The Properties screen of the monitor opens.

To delete a monitor

1. On the Main tab of the navigation pane, expand **Link Controller** **Local Traffic** then click **Monitors**.
The Monitor List screen opens.
2. Click the Select box for the monitor that you want to delete.
3. Click the **Delete** button.
The Delete confirmation message displays.
4. Click the **Delete** button.

To enable or disable a monitor instance

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Monitors**.
The Monitor List screen opens.
2. Click a monitor name in the list.
3. On the menu bar, click **Instances**.
4. For the instance you want to manage, click the Select box.
5. Click either the **Enable** or **Disable** button.



9

Inbound Load Balancing

- Introducing inbound load balancing
- Understanding inbound load balancing on the Link Controller
- Using static load balancing modes
- Using dynamic load balancing modes
- Configuring inbound load balancing

Introducing inbound load balancing

The Link Controller™ system uses different configuration modes to manage traffic across multiple links. As requests come across the Link Controller, the system identifies the best available virtual server in the Link Controller system configuration and sends the traffic to that virtual server. This process is known as ***inbound load balancing***.

When you work with inbound load balancing and the Link Controller system, it is important to:

- Understand how inbound load balancing works in a Link Controller system context
- Understand static load balancing modes and how they apply to the needs of your network traffic
- Understand dynamic load balancing modes and how they apply to the needs of your network traffic
- Configure the Link Controller system to use the load balancing modes that best apply to your network

Understanding inbound load balancing on the Link Controller

The Link Controller system has external IP addresses configured for each network or ISP link that it manages. The external IP addresses are configured as virtual servers on the Link Controller system, and these virtual servers belong to a wide IP on the Link Controller system. The translations for the virtual servers direct the incoming requests to the appropriate content servers (for example, a web server or a database server) on the internal network. This process is known as ***inbound load balancing***.

When the Link Controller system receives a name resolution request from a local DNS server, the system uses a load balancing mode to select the best available virtual server in a wide IP. Once the Link Controller system selects the best virtual server, it constructs the DNS answer and sends the answer back to the requesting client's local DNS server, using the best available link. The DNS answer, or ***resource record***, is an **A** record that contains one or more virtual server IP addresses.

The Link Controller system chooses a virtual server from a wide IP using either a static load balancing mode or a dynamic load balancing mode. A ***static load balancing mode*** selects a virtual server based on a pre-defined pattern. A ***dynamic load balancing mode*** selects a virtual server based on current performance metrics.

Within each wide IP, you can specify up to three load balancing modes that the system uses in sequential order: the preferred method, the alternate method, and the fallback method. The ***preferred*** method is the first load balancing mode that the Link Controller system uses for load balancing. If

the preferred method fails to select a virtual server, the system then uses the alternate method for load balancing. If the alternate load balancing mode fails to select a virtual server, the system uses the fallback load balancing mode. If the fallback method fails, then the Link Controller system randomly picks an available virtual server.

The Link Controller system supports several modes for inbound load balancing. Table 9.1 shows a complete list of the load balancing modes, indicates where you can use each mode, and indicates whether the mode is static or dynamic. The sections following the table describe how each load balancing mode works.

Load Balancing mode	Use for preferred method	Use for alternate method	Use for fallback method
Completion Rate	X		X
Global Availability	X	X	X
Hops	X		X
Kilobytes/Second	X		X
Least Connections	X		X
Packet Rate	X	X	X
Quality of Service	X		X
Ratio	X	X	X
Round Robin	X	X	X
Round Trip Time	X		X
Static Persist	X	X	X
Topology	X	X	X
Fallback IP	X	X	X
Drop Packet	X	X	X
Connection Rate	X	X	X
VS Capacity	X	X	X
Virtual Server Score	X	X	X

Table 9.1 Load balancing mode usage

Static load balancing modes distribute requests according to predefined patterns, and take virtual server availability into account. *Dynamic load balancing modes* distribute requests to links that show the best current performance. The performance metrics taken into account depend on the particular dynamic mode you are using.

Using static load balancing modes

Static load balancing modes distribute connections according to predefined patterns, and take server availability into account. The Link Controller system supports the following static load balancing modes:

- Drop Packet
- Fallback IP
- Global Availability
- Ratio
- Round Robin
- Static Persist
- Topology

The None load balancing mode is a special mode that you can use to skip load balancing under certain conditions. The other static load balancing modes perform true load balancing as described in the following sections.

Drop Packet mode

When you specify the Drop Packet load balancing mode, the Link Controller system does nothing with the packet, and simply drops the request.

◆ **Note**

A typical LDNS server iteratively queries other authoritative name servers when it times out on a query.

We recommend that you use the Drop Packet load balancing mode only for the fallback method. The Link Controller system uses the fallback method when the preferred and alternate load balancing modes do not provide at least one virtual server to return as an answer to a query.

Fallback IP mode

When you specify the Fallback IP mode, the Link Controller system returns the IP address that you specify as the fallback IP as an answer to the query. Note that the IP address that you specify is not monitored for availability

before being returned as an answer. When you use the Fallback IP mode, you can specify a disaster recovery site to return when no load balancing mode returns an available virtual server. We recommend that you use the Fallback IP load balancing mode only for the fallback method. The Link Controller system uses the fallback method when the preferred and alternate load balancing modes do not provide at least one virtual server to return as an answer to a query.

Global Availability mode

The Global Availability load balancing mode uses the virtual servers included in the pool in the order in which they are listed. For each connection request, this mode starts at the top of the list and sends the connection to the first available virtual server in the list. Only when the current virtual server is full or otherwise unavailable does Global Availability mode move to the next virtual server in the list. Over time, the first virtual server in the list receives the most connections and the last virtual server in the list receives the least number of connections.

Ratio mode

The Ratio load balancing mode distributes connections among a pool of virtual servers as a weighted Round Robin. For example, you can configure the Ratio mode to send twice as many connections to a fast, new server, and only half as many connections to an older, slower server.

The Ratio load balancing mode requires that you define a ratio weight for each virtual server in a pool, or for each pool if you are load balancing requests among multiple pools. The default ratio weight for a server or a pool is set to 1.

Round Robin mode

The Round Robin load balancing mode distributes connections in a circular and sequential pattern among the virtual servers in a pool. Over time, each virtual server receives an equal number of connections.

Static Persist mode

The Static Persist load balancing mode provides static persistence of local DNS servers to virtual servers; it consistently maps an LDNS IP address to the same available virtual server for the duration of the session. This mode guarantees that certain transactions are routed through a single transaction manager (for example, a Local Traffic Manager™ system or other server array manager); this is beneficial for transaction-oriented traffic, such as e-commerce shopping carts, online trading, and online banking.

Topology mode

The Topology load balancing mode allows you to direct or restrict traffic flow by adding topology records to a topology statement in the configuration file. When you use the Topology load balancing mode, you can develop proximity-based load balancing. For example, a client request from a particular IP subnet can be directed to a specific IP subnet on your network. The Link Controller system determines the proximity of servers by comparing location information derived from the DNS message to the topology records.

This load balancing mode requires you to do some advanced configuration planning, such as gathering the information you need to define the topology records. See Chapter 10, *Working with Topologies*, for detailed information about working with this and other topology features.

Using dynamic load balancing modes

Dynamic load balancing modes distribute connections to links that show the best current performance. The performance metrics taken into account depend on the particular dynamic mode you are using.

Types of dynamic load balancing modes

The Link Controller system supports the following dynamic load balancing modes:

- Connection Rate
- Completion Rate
- Hops
- Kilobytes/Second
- Least Connections
- Packet Rate
- Round Trip Times (RTT)
- Quality of Service (QOS)
- VS Capacity

Connection Rate mode

The Connection Rate mode selects the virtual server that is currently accepting the fewest number of connections.

Completion Rate mode

The Completion Rate load balancing mode selects the virtual server that currently maintains the least number of dropped or timed-out packets during a transaction between a data center and the client LDNS.

Hops mode

The Hops load balancing mode is based on the **traceroute** utility, and tracks the number of intermediate system transitions (router hops) between a client LDNS and each data center. Hops mode selects a virtual server in the data center that has the fewest router hops from the LDNS server.

Kilobyte/Second mode

The Kilobytes/Second load balancing mode selects a virtual server that is currently processing the fewest number of kilobytes per second.

◆ Note

You can use the Kilobytes/Second mode only with servers for which the Link Controller system can collect the kilobytes per second metric.

Least Connections mode

The Least Connections load balancing mode is used for load balancing to virtual servers managed by a load balancing server, such as a Local Traffic Manager. The Least Connections mode simply selects a virtual server on the Local Traffic Manager system that currently hosts the fewest connections.

Packet Rate mode

The Packet Rate load balancing mode selects a virtual server that is currently processing the fewest number of packets per second.

Round Trip Times mode

The Round Trip Times (RTT) load balancing mode selects the virtual server with the fastest measured round trip time between a data center and a client LDNS.

Quality of Service mode

The Quality of Service load balancing mode uses current performance information to calculate an overall score for each virtual server, and then distributes connections based on each virtual server's score. The performance factors that the Link Controller system takes into account include:

- Round trip time
- Hops
- Connection rate
- Packet rate
- Topology
- Link Capacity
- VS Capacity
- Kilobytes/Second

The Quality of Service load balancing mode is a customizable load balancing mode. For simple configurations, you can easily use this load balancing mode with its default settings. For more advanced configurations, you can specify different weights for each performance factor in the equation.

You can also configure the Quality of Service load balancing mode to use the dynamic ratio feature. With the dynamic ratio feature turned on, the Quality of Service mode becomes similar to the Ratio mode, where the connections are distributed in proportion to ratio weights assigned to each virtual server. The ratio weights are based on the QOS scores: the better the score, the higher percentage of connections the virtual server receives.

For details about customizing the Quality of Service mode, see the *Customizing the QOS equation*, on page 9-10.

VS Capacity mode

The VS Capacity load balancing mode creates a list of the virtual servers, weighted by capacity, then picks one of the virtual servers from the list. The virtual servers with the greatest capacity are picked most often, but over time all virtual servers are returned. If more than one virtual server has the same capacity, then the Link Controller system load balances using the Round Robin mode among those virtual servers.

Virtual Server Score

The Virtual Server Score load balancing mode instructs the Global Traffic Manager™ system to assign connection requests to virtual servers based on a user-defined ranking system. This load balancing mode is only available for managing connections between virtual servers controlled through BIG-IP® Local Traffic Manager™ systems.

Unlike other settings that affect load balancing operations, you cannot assign a virtual server score to a virtual server through the Global Traffic Manager system. Instead, you assign this setting through the Local Traffic Manager responsible for the virtual server. See the *Configuration Guide for BIG-IP® Local Traffic Manager™* for more information.

Implementing the Quality of Service load balancing mode

The Quality of Service mode is a dynamic load balancing mode that includes a configurable combination of the Round Trip Time (RTT), Completion Rate, Packet Rate, Topology, Hops, Link Capacity, VS Capacity, and Kilobytes/Second (KBPS) modes. The Quality of Service mode is based on an equation that takes each of these performance factors into account. When the Link Controller system selects a virtual server, it chooses the server with the best overall score.

The Quality of Service mode has default settings that make it easy to use: simply specify Quality of Service as your preferred load balancing mode. There is no need to configure Quality of Service, but if you want to change the settings, you can customize the equation to put more or less weight on each individual factor. The following topics explain how to use and adjust the various settings.

Understanding QOS coefficients

Table 9.2 lists each Quality of Service (QOS) coefficient, its scale, a likely upper limit for each, and whether a higher or lower value is more efficient.

Coefficient	How measured	Default value	Example upper limit	Higher or lower?
Packet rate	Packets per second	1	700	Lower
Round trip time	Microseconds	50	2,000,000	Lower
Hit Ratio	Percentage of successfully transferred packets (0-100%)	5	100%	Higher
Topology	Score that defines network proximity by comparing server and LDNS IP addresses (0-2 ³²)	0	100	Higher
Hops	Number of intermediate systems transitions (hops)	0	64	Lower
Link Capacity	Bandwidth usage	30	2,000,000	Higher
VS capacity	Number of nodes up	0	20	Higher
Connection Rate	Percentage of connections made	0	100	Lower
Kilobytes/second	Kilobytes per second throughput	3	15000	Lower

Table 9.2 QOS coefficients: Default values, ranges, and limits

If you change the default QOS coefficients, keep the following issues in mind.

- ◆ **Scale**

The raw metrics for each coefficient are not on the same scale. For example, completion rate is measured in percentages, while the packet rate is measured in packets per second.

- ◆ **Normalization**

The Link Controller system normalizes the raw metrics to values in the range of 0 to 10. As the QOS value is calculated, a high measurement for completion rate is good, because a high percentage of completed

connections are being made, but a high value for packet rate is not desirable because the packet rate load balancing mode attempts to find a virtual server that is not overly taxed at the moment.

- **Emphasis**

You can adjust coefficients to emphasize one normalized metric over another. For example, consider the following QOS configuration:

- Round Trip Time: 50
- Hops: 0
- Topology: 0
- Hit Ratio: 5
- Packet Rate: 1
- VS Capacity: 0
- Bits/second: 3
- Link Capacity: 30
- Connection Rate: 0

In this configuration, if the completion rates for two virtual servers are close, the virtual server with the best packet rate is chosen. If both the completion rates and the packet rates are close, the round trip time (RTT) breaks the tie. In this example, the metrics for Topology, Hops, Link Capacity, VS Capacity, and Kilobytes/Second modes are not used in determining how to distribute connections.

◆ **Note**

*You cannot set a value for both the **Round Trip Time** and **Hops** settings simultaneously. In situations where the Link Controller system has a value for both settings, the **Round Trip Time** value is incorporated, while the value for the **Hops** setting is reset to 0.*

Customizing the QOS equation

If you want to establish your own custom settings for the Quality of Service load balancing method, you can do so at any time. You can only customize the Quality of Service equation at the pool level.

To customize the QOS equation

1. On the Main tab in the navigation pane, expand **Link Controller** and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the name of the wide IP for which you want to modify the QOS equation.
The properties screen for that wide IP opens.
3. On the menu bar, click **Members**.
The members screen opens.

4. From either the **Preferred** or **Fallback** list, select **Quality of Service**.
5. Define the global QOS coefficients in the appropriate fields.
6. Click the **Update** button to save your changes.

Using the Dynamic Ratio option

The dynamic load balancing modes also support the **Dynamic Ratio** option. When you activate this option, the Link Controller system treats dynamic load balancing values as ratios, and it uses each server in proportion to the ratio determined by this option. When the **Dynamic Ratio** option is off, the Link Controller system uses only the server with the best result based on the dynamic load balancing mode you implemented (in which case it is a winner-takes-all situation), until the metrics information is refreshed.

◆ Note

*By default, the **Dynamic Ratio** option is off.*

To illustrate how the **Dynamic Ratio** option works, consider a pool, **primaryOne**, that contains several pool members. This pool is configured so that the Link Controller system load balances name resolution requests based on the Round Trip Time load balancing mode. The **primaryOne** pool contains two pool members: **memberOne** and **memberTwo**. For this example, the Link Controller system determines that the round trip time for **memberOne** is 50 microseconds, while the round trip time for **memberTwo** is 100 microseconds.

If the **primaryOne** pool has the **Dynamic Ratio** option disabled (the default setting), the Link Controller system always load balances to the pool with the best value. In this case, this results in requests going to **memberOne**, because it has the lowest round trip time value.

If the **primaryOne** pool has the **Dynamic Ratio** option enabled, however, the Link Controller system treats the round trip time values as ratios and divide requests among pool members based on these ratios. In this case, this results in **memberOne** getting twice as many connections as **memberTwo**, because the round trip time for **memberOne** is twice as fast as the round trip time for **memberTwo**. Note that, with the **Dynamic Ratio** option enabled, both pool members are employed to handle connections, while if the option is disabled, only one pool member receives connections.

To turn on the Dynamic Ratio option

1. On the Main tab in the navigation pane, expand **Link Controller** and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the name of the wide IP for which you want to enable the **Dynamic Ratio** option.
The properties screen for the wide IP opens.

3. From the **Configuration** list, select **Advanced**.
4. Check the **Dynamic Ratio** check box
5. Click the **Update** button to save your changes.

Configuring inbound load balancing

Once you identify which load balancing methods are best for your network, you can configure the Link Controller system to implement those methods. This process ensures that traffic flows through your network as efficiently as possible. You configure inbound load balancing at the wide IP level.

When you define a wide IP, you specify the preferred and alternate load balancing methods to use in selecting a virtual server within the wide IP.

To configure inbound load balancing

1. On the Main tab in the navigation pane, expand **Link Controller** and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the name of the wide IP to which you want to configure load balancing.
The properties screen for the wide IP opens.
3. On the menu bar, click **Members**.
4. For the **Load Balancing Method** setting, select the appropriate preferred, alternate, and fallback load balancing methods.
5. Click the **Update** button to save your changes.

Changing the load balancing methods

By default, the preferred load balancing method is set to Quality of Service, and the alternate and fallback load balancing methods are set to Round Robin. For details on these load balancing modes as well as the additional load balancing modes, see *Introducing inbound load balancing*, on page 9-1.

Using the Round Robin LDNS wide IP attribute

Round Robin LDNS is an attribute that you can use in conjunction with any load balancing mode to return a list of available virtual servers, instead of a single virtual server. Certain browsers keep the answer returned by DNS servers. By enabling this attribute, the Link Controller system returns a maximum of 16 virtual servers as the answer to a DNS resolution request. This provides browsers with alternate answers if a virtual server becomes unavailable.

Adjusting the QOS coefficients

You can adjust the QOS coefficients to best match your load balancing requirements. By default, only the Link Capacity coefficient has a value. For more information on working with the QOS coefficients, see *Understanding QOS coefficients*, on page 9-9.



10

Working with Topologies

- Introducing topologies
- Setting up and removing topology records
- Using topology load balancing in a wide IP
- Understanding user-defined regions
- Other load balancing options for topologies

Introducing topologies

The Link Controller™ system handles name resolution requests at an international level. Consequently, one of the methods you can employ to load balance requests is through the use of topologies. A **topology** is a set of characteristics that identifies the origin of a given name resolution request. In the Link Controller system, topologies belong to one of several categories, including:

- IP Subnet
- ISP
- Region

A **region** is a customized collection of topologies. For example, you could create a topology for each of a number of ISPs in Denmark, Iceland, Finland, Norway, and Sweden. These topologies could then compose a custom region called Scandinavia.

Through topologies, you can instruct the Link Controller system to select a resource based on its physical proximity to the client making the name resolution request. This process helps ensure that name resolution requests are answered and managed in the fastest possible time.

You can instruct the Link Controller system to use topologies to load balance name resolution requests across virtual servers at the wide IP level.

Understanding topologies

A fictional company, SiteRequest, allows its customers to download applications from its web site. SiteRequest has three major customers, one each in New York, Paris, and Tokyo. To ensure that customers can download their purchased application as quickly as possible, the IT department has decided to create topologies with which to load balance name resolution requests.

SiteRequest has created a custom region named Paris to handle the name resolution requests originating in Paris. SiteRequest has assigned the IP subnet **192.168.15.2** of the SiteRequest network as the destination for the requests from Paris. With this custom region created, the next step is to create a topology record for the Link Controller system. A **topology record** is a statement that tells the system how to handle name resolution requests based on topologies. In this case, the IT department creates the record as follows:

- Request Source: Region is Paris
- Destination Source: IP subnet is **192.168.15.2**
- Weight: 10

The final step to implement this topology is to configure the corresponding wide IP, **www.siterequest.com**, to use topology load balancing. See *Using topology load balancing in a wide IP*, on page 10-5, for more information.

Implementing topologies

When you want to load balance connection requests using one or more topologies, you must complete two tasks:

- Configure the given wide IP or pool to use topology as a load balancing method.
- Access the Topology screen to create your topology statements.

To configure a wide IP or pool to use topology as a load balancing method, see *Configuring inbound load balancing*, on page 9-13.

To access the topology screen

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Topology**.
The Topology Records screen opens.
2. Create and manage your topology statements as needed.

For more information, see *Setting up and removing topology records*, following.

Setting up and removing topology records

A topology record has several elements: a request source statement, a destination statement, an operator, and a weight.

A **request source statement** defines the origin of a name resolution request. You can define the origin of a request as one of the following:

- An IP subnet (CIDR definition)
- An Internet Service Provider (ISP)
- A custom region

A **destination statement** defines the resource to which the Link Controller system directs the name resolution request. The types of resources available for a destination statement are as follows:

- An IP subnet (CDIR definition)
- An Internet Service Provider (ISP)
- A custom region

You can select one of two operators for both a request source and a destination statement. The **is** operator indicates that the name resolution request matches the statement. The **is not** operator indicates that the name resolution request does not match the statement.

The last element of a topology record, called the topology score or **weight**, allows the Link Controller system to evaluate the best resolution option for a DNS request. In the event that a name resolution request matches more than one topology record, the Link Controller system uses the record with the highest weight attribute to determine which statement it uses to load balance the request.

◆ Note

*A group of topology records defined for the Link Controller system is referred to as a **topology statement**.*

To set up a topology record

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Topology**.
The Topology Records screen opens.
2. Click the **Create** button.
The New Record screen opens.

3. To create a request source statement, use the **Request Source** setting:
 - a) Select an origin type from the **Request Source** list.
 - b) Select an operator, either **is** or **is not**.
 - c) Define the criteria for the request source statement. For example, if the statement focuses on an IP subnet, a box displays that allows you to define that subnet.
4. To create a destination statement, use the **Destination** setting:
 - a) Select a destination type from the **Destination** list.
 - b) Select an operator, either **is** or **is not**.
 - c) Define the criteria for the destination statement. For example, if the statement focuses on an IP subnet, a box displays that allows you to define that subnet.
5. In the **Weight** box, specify the priority this record has over topology records.
6. Click the **Create** button to save the new topology.

Removing topology records

As your network changes, you might find that you need to refine your existing topology records, or remove outdated topology records.

For example, the fictional company SiteRequest has an existing topology statement that routes all traffic originating from the United States to the New York data center. Last week, a new data center in Los Angeles came online. One of the results of this new data center is that the topology record that the Link Controller system used to direct traffic was obsolete, and needed to be removed.

To remove a topology record

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Topology**.
The Topology Records screen opens.
2. Select the topology record that you want to remove from the topology records list by checking the corresponding Select check box.
3. Click the **Delete** button.

Using topology load balancing in a wide IP

You can use the Topology load balancing mode to distribute traffic among the pools in a wide IP. To do this, you must have at least two pools configured in the wide IP. With topology load balancing, you send name resolution requests to specific resources based on the origin of the request.

To configure a wide IP to use topology load balancing

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Inbound Wide IPs**.
The Wide IP List screen opens.
2. Click the name of the wide IP for which you want to assign topology-based load balancing.
The properties screen for the wide IP opens.
3. On the menu bar, click **Members**.
The Members screen opens. This screen contains a list of the pools currently assigned to the wide IP.
4. For the **Load Balancing Method** setting, select **Topology** as the preferred, alternate, or fallback method.
5. Click the **Update** button to save your changes.

Repeat this process for each wide IP as needed.

Understanding user-defined regions

To further refine the topology load balancing capabilities of the Link Controller system, you can create custom topology regions. A **region** is a customized collection of topologies. For example, you could create a topology for Denmark, Iceland, Finland, Norway, and Sweden. These topologies could then compose a custom region for Scandinavia. Regions allow you to extend the functionality of your topologies by allowing you to define specific geographical regions that have meaning for your network.

You create a custom region by adding one or more region member types to the region member list. The available region member types are as follows:

- An IP subnet (CDIR definition)
- An Internet Service Provider (ISP)
- Another custom region

Once you select a region member type, you then fill in the details about that region member and add it to the region member list. The region member options change based on the region member type that you select. When you have finished adding region members to your new region, the new region becomes an option in the Create Topology screen.

To create a region

1. On the Main tab of the navigation pane, expand **Link Controller** and then click **Topology**.
The Topology Records screen opens.
2. On the menu bar, click **Regions**.
The main region screen opens.
3. Click the **Create** button.
The Create Region screen opens.
4. In the **Name** box, type a name for the new region.
5. Using the **Member List** setting, define the appropriate region members.
6. Click the **Create** button to create the new region.

Other load balancing options for topologies

The Link Controller system supports the **Longest Match** option that affects how the system load balances name resolution requests.

The **Longest Match** option instructs the Link Controller system to use the topology statement that most completely matches the source IP address of the name resolution request. For example, two topology statements exist: one that matches a source IP address of **10.0.0.0** and one that matches **10.15.0.0**. A name resolution request arrives with a source IP address of **10.15.65.8**. With the Longest Match setting enabled, the Link Controller system uses the topology statement with **10.15.0.0** because it has the longest, and therefore most complete, match. If this option is disabled, the system uses either topology statement, depending on factors such as the weight of the statement or the order in which the statements are listed. This option is enabled by default.

To access the Longest Match option

1. On the Main tab of the navigation pane, expand **System** and then click **Configuration**.
The General Properties screen opens.
2. From the Global Traffic menu, choose Load Balancing.
The load balancing properties screen opens.
3. In the Topology Options section, check or clear the **Longest Match** option as needed.
4. Click the **Update** button to save your changes.



II

Synchronizing Link Controllers

- Introducing synchronization
- Activating synchronization
- Controlling file synchronization
- Creating synchronization groups
- Running the gtm_add script
- Synchronizing Link Controller and Global Traffic Manager systems

Introducing synchronization

The primary goal of the Link Controller™ system is to ensure that network traffic flows as efficiently and as cost-effectively as possible. Often, Link Controller systems are installed in pairs on a given network segment, ensuring that if one system should fail, another is available to prevent network downtime. In environments where two Link Controller systems are on the same subnet, you can configure these systems so that a change made to one Link Controller system automatically transfers to the other. This process is called synchronization.

In network configurations that contain more than one Link Controller system, **synchronization** means that each system regularly compares the timestamps of its configuration files with each other. If, at any time, a system discovers that its configuration files are older than the configuration files on another Link Controller system, it automatically transfers the newest configuration files to itself. With synchronization, you can change settings on one system and have that change distributed to all other systems.

You can separate the Link Controller systems on your network into separate groups, called synchronization groups. A **synchronization group** is a collection of multiple Link Controller systems that share and synchronize configuration settings. These groups are identified by a synchronization group name, and only systems that share this name share configuration settings. These synchronization groups allow you to customize the synchronization behavior. For example, the Link Controller systems residing in data centers in Europe might belong to one synchronization group, while the systems in North America belong to another group.

One exception to this process occurs when you add an additional Link Controller system to the network. In this scenario, there is a chance that the timestamp of the configuration file on the system you are adding to the network is newer than the configuration files on the already-installed Link Controller systems. When you add the new Link Controller system to the network, if you enable synchronization on the new system, the configuration file on the new system is distributed to the existing Link Controller systems, effectively removing the existing configurations. Therefore, you should run the **gtm_add** script on the new Link Controller system before you enable synchronization on that system.

The **gtm_add** script acquires the configuration file from an existing Link Controller system and applies it to the new system. As a result, the new system has the current configuration for your network.

You interact with the Link Controller system's synchronization settings in a variety of ways. You can:

- Define NTP servers
- Activate synchronization
- Control file synchronization, including running the **gtm_add** script when you add a new Link Controller system to your network

- Synchronize DNS zone files
- Create synchronization groups

◆ **Note**

*If your network includes both Link Controller systems and Global Traffic Manager™ systems, we highly recommend you review the section, **Synchronizing Link Controller and Global Traffic Manager systems**, on page 11-6, to ensure that you understand how these two products handle synchronization when they share the same network.*

Defining NTP servers

Before you can synchronize Link Controller systems, you must define the Network Time Protocol (NTP) servers that the systems reference. These servers ensure that the each Link Controller system is referencing the same time when verifying timestamps for configuration files.

To define an NTP server

1. On the Main tab of the navigation pane, expand **System** and then click **Configuration**.
The General Properties screen opens.
2. From the Device menu, choose NTP.
The NTP screen opens.
3. In the **Address** box, type either the IP address or fully-qualified domain name for the time server.
4. Click the **Add** button to add the NTP server to your configuration.
The time server displays as an entry in the **Time Server List**.
5. Click **Update** to save your changes.

Repeat this process for any additional time servers.

Activating synchronization

Activating synchronization for the Link Controller system has an immediate effect on its configurations, provided that another Link Controller system is already available on the network. We recommend that you activate synchronization after you have finished configuring one of the systems.

To activate synchronization

1. On the Main tab of the navigation pane, expand **System** and then click **Configuration**.
The General Properties screen opens.
2. From the Global Traffic menu, choose General.
The general global properties screen opens.
3. Check the **Synchronization** check box.
4. Click the **Update** button to save your changes.

Controlling file synchronization

When you opt to synchronize multiple Link Controller systems, you are instructing each system to share its configuration files with the other systems on the network. These files are synchronized based on their timestamp: if a Link Controller system determines that its configuration files are older than those at another system, it acquires the newer files and begins using them to load balance name resolution requests.

You can control the synchronization by defining the maximum age difference between two sets of configuration files. This value is referred to as synchronization time tolerance.

By default, the value for the synchronization time tolerance is set to 10 seconds. The minimum value you can set for this value is 5 seconds, while the maximum you can set is 600 seconds.

To control file synchronization

1. On the Main tab of the navigation pane, expand **System** and then click **Configuration**.
The General Properties screen opens.
2. From the Global Traffic menu, choose General.
The general global properties screen opens.
3. In the **Synchronization Time Tolerance** box, type the maximum age difference, in seconds, between two sets of configuration files.
4. Click the **Update** button to save your changes.

Deactivating file synchronization

In the event that you need to deactivate file synchronization, you can do so at any time. Situations in which you want to disable synchronization include updating the data center in which the Link Controller system resides, or when you are testing a new configuration change.

To deactivate file synchronization

1. On the Main tab of the navigation pane, expand **System** and then click **Configuration**.
The General Properties screen opens.
2. From the Global Traffic menu, choose General.
The general global properties screen opens.
3. Clear the **Synchronization** check box.
4. Click the **Update** button to save your changes.

Creating synchronization groups

Each Link Controller system that you synchronize must belong to a specific group of systems, called a synchronization group. A **synchronization group** is a collection of multiple Link Controller systems that share and synchronize configuration settings. Initially, when you enable synchronization for a Link Controller, the system belongs to a synchronization group called Default. However, you can create new groups at any time. This process allows you to customize the synchronization process, ensuring that only certain sets of Link Controller systems share configuration values.

To illustrate how synchronization groups work, consider the fictional company, SiteRequest. SiteRequest has decided to add a new data center in Los Angeles. As part of bringing this data center online, SiteRequest has decided that it wants the Link Controller systems installed in New York and in Los Angeles to share configurations, and the Paris and Tokyo data centers to share configurations. This setup exists because SiteRequest's network optimization processes require slightly different settings within the United States than the rest of the world. To accommodate this new network configuration, SiteRequest enables synchronization for the New York and Los Angeles data centers, and assigns them a synchronization group name of United States. The remaining data centers are also synchronized, but with

a group name of Rest Of World. As a result, a configuration change at the Paris Link Controller system immediately modifies the Tokyo system, but does not affect the systems in the United States.

◆ **Note**

When you change the synchronization group name for a group, you must manually change it for each system within the synchronization group, as this value does not synchronize. In addition, synchronization stops for any systems with names that do not match.

To create a synchronization group

1. On the Main tab of the navigation pane, expand **System** and then click **Configuration**
The General Properties screen opens.
2. From the Global Traffic menu, choose General.
The general global properties screen opens.
3. In the **Synchronization Group Name** box, type a name of either an existing synchronization group, or a new group.

***Important:** If your network includes Global Traffic Manager systems as well as Link Controller systems, they **cannot** share the same synchronization name. Doing so could cause unintended results.*

4. Click the **Update** button to save your changes.

Running the gtm_add script

When you add an additional Link Controller system to a network in which a Link Controller system already exists, you must run the **gtm_add** script on the new system. You run the **gtm_add** script to pull the configuration from the existing system to the new system.

To run the gtm_add script

1. Log on to the new Link Controller system that you added to the network.
2. At the command prompt, type **gtm_add**.
A prompt displays, describing what the **gtm_add** script does and asking if you are sure you want to run the process.
3. Press the **y** key to start the **gtm_add** script.
The script prompts you for the IP address of the system from which you want to acquire configuration settings.
4. Type the IP address of the configured Link Controller system.
5. Press **Enter**.

At this point, both Link Controller systems share the same configuration. In addition, they also belong to the same synchronization group, because the **gtm_add** script copied the settings from the existing Link Controller system to the new Link Controller system.

Synchronizing Link Controller and Global Traffic Manager systems

It is possible for a network to contain both Link Controller systems and Global Traffic Manager systems. You must take care when implementing synchronization in such network configurations, because the synchronization feature treats both Link Controller systems and Global Traffic Manager systems as the same. However, the Global Traffic Manager system has a larger set of responsibilities than the Link Controller system. As a result, if you do not use caution when implementing synchronization, you could configure a Global Traffic Manager system in a way that is invalid for a Link Controller system. This issue is especially relevant when configuring wide IPs, as the Global Traffic Manager system works with wide IPs at a more detailed level than the Link Controller system does.

When implementing synchronization in a network that has both Link Controller systems and Global Traffic Manager systems, we recommend that you remember the following:

- The synchronization group name for the Link Controller systems should be different than the synchronization group name for the Global Traffic Manager systems. This setup ensures that the Link Controller systems are synchronized separately from changes made to any Global Traffic Manager systems on the network.
- If, through synchronization, a Link Controller system receives a wide IP configuration that it cannot resolve, that wide IP displays in the system with a gray status code. This code indicates that the system knows of the wide IP, but does not have the ability to modify it.



12

Viewing Statistics

- Introducing statistics
- Accessing statistics
- Understanding the types of statistics

Introducing statistics

One of the most important aspects to managing a network is timely access to accurate information on network performance. This information can verify that the Link Controller™ system is handling your name resolution requests as efficiently as possible, as well as provide data on the overall performance of a specific resource, such as a data center or distributed application.

The Link Controller system gathers statistical data on multiple aspects of your network. You access these statistics through the statistics screen. The types of statistics you can select from this screen include:

- Wide IPs
- Wide IP members
- Links
- Paths
- Local DNS

The Link Controller system gathers statistics through a software component called the **big3d** agent. This agent is responsible for managing the various monitors that you assign to your network components, and returning statistics based on those monitors back to the Link Controller system.

Accessing statistics

You can access Link Controller system statistics in two ways:

- Through the Statistics option on the Main tab of the navigation pane
- Through the Statistics menu from various main screens for different components

Both methods open the same screen within the Link Controller system. When you access statistics through a menu on the main screen for a given network component, the Statistics screen is pre-configured for the given network element, although you can switch to a different set of statistics at any time.

To access statistics through the Main tab

1. On the Main tab of the navigation pane, expand **Overview** and then click **Statistics**.
The Statistics screen opens.
2. From the **Statistics Type** list, select the type of statistics you want to view.
These statistics are described in later sections of this chapter.

3. Select the data format in which you want to view the statistics:
 - If you select **Normalized**, the Link Controller system rounds the data to the nearest digit.
 - If you select **Unformatted**, the Link Controller system displays the exact value to as many decimal places as the value requires.
4. From the **Auto Refresh** list, select the frequency at which the Link Controller system refreshes data on the screen.
If you select **Disabled** from this list, the system does not refresh the screen; instead, you can click the **Refresh** button to update the screen with the latest statistical data.

To access statistics through a component's main screen

1. On the Main tab of the navigation pane, expand **Link Controller** and click a component, such as **Links**.
The main screen for the component opens.
2. On the menu bar, click **Statistics**.
The Statistics screen opens. This screen is pre-configured to display statistics relevant to the component.
3. Select the data format in which you want to view the statistics:
 - If you select **Normalized**, the Link Controller system rounds the data to the nearest digit.
 - If you select **Unformatted**, the Link Controller system displays the exact value to as many decimal places as the value requires.
4. From the **Auto Refresh** list, select the frequency at which the Link Controller system refreshes data on the screen.
If you select **Disabled** from this list, the system does not refresh the screen; instead, you can click the **Refresh** button to update the screen with the latest statistical data.

Understanding the types of statistics

You can view a variety of statistics through the Link Controller system, including:

- ◆ **Wide IPs**
The statistics for wide IPs provide you with information on what wide IPs exist and how the Link Controller system has load balanced traffic to the wide IP.
- ◆ **Wide IP members**
The statistics for wide IP members provide details on the resources that are a part of a specific wide IP.
- ◆ **Links**
The statistics for links focus on how much traffic is flowing in and out through a specific link to the Internet.
- ◆ **Paths**
The statistics for paths provide information on how quickly traffic moves between a Local DNS and a resource for which the Link Controller system is responsible.
- ◆ **Local DNS**
The statistics for local DNS servers provide location details related to the different Local DNS servers that communicate with the Link Controller system.

Wide IP statistics

The Link Controller system captures several statistics related to the performance of a wide IP. These statistics primarily focus on how many resolution requests have been sent for the wide IP, and how the Link Controller system has load balanced these requests. You can access the wide IP statistics by selecting **Wide IPs** from the **Statistics Type** list in the Statistics screen. For information on accessing the Statistics screen, see *Accessing statistics*, on page 12-1.

As an example of wide IP statistics, consider the fictional company SiteRequest. The IT department at SiteRequest has a wide IP, **www.siterequest.com**, which uses the Global Availability load balancing mode. This mode sends all name resolution requests for this wide IP to a specific pool until that pool is unavailable. Because the wide IP, **www.siterequest.com**, is critical to SiteRequest's operations, the IT department wants to track traffic to the wide IP and ensure that the primary pool is not at risk of getting overloaded. The wide IP statistics provide the IT department the information they need to see how many requests are being sent for the wide IP, allowing them to plan additional resource allocations more effectively.

The wide IP statistics screen consists of a Wide IP Statistics table. This table contains the following information:

◆ **Status**

The Status column indicates the current status of the wide IP. The available status types are: Available, Unavailable, Offline, and Unknown. Each status type is represented by a symbol; for example, the available status type is represented by a green circle.

◆ **Wide IP**

The Wide IP column displays the name of a wide IP for which the Link Controller system is responsible. Each name displays as a link. When you click the link, the properties screen for the wide IP opens.

◆ **Members**

The Members column provides a link that opens a member details screen for the wide IP. This screen displays load balancing statistics for each resource within the wide IP. You can return to the main wide IP statistics screen by clicking the **Back** button in the Display Options area of the screen.

◆ **Requests**

The Requests column displays the cumulative number of DNS requests sent to the wide IP.

◆ **Requests Persisted**

The Requests Persisted column displays the cumulative number of requests that persisted. Persisted requests use the same pool during a connection session.

◆ **Load Balancing**

The Load Balancing column provides information on how the Link Controller system load balanced connection requests to this resource. This column consists of four subcolumns:

- The Preferred subcolumn displays the cumulative number of requests that the Link Controller system load balanced with the preferred load balancing method.
- The Alternate subcolumn displays the cumulative number of requests that the Link Controller system load balanced with the alternate load balancing method.
- The Fallback subcolumn displays the cumulative number of requests that the Link Controller system load balanced with the Fallback load balancing method.
- The Returned to DNS subcolumn displays the cumulative number of requests that the Link Controller system could not resolve and returned to the Domain Name Server (DNS).

Wide IP member statistics

Wide IP member statistics provide information on the amount of traffic flowing to and from each resource that belongs to a specific wide IP. This information can tell you if your resources are distributed appropriately for your network. You can access the virtual server statistics by selecting **Wide IP Members** from the **Statistics Type** list in the Statistics screen.

As an example of how the statistics for servers can help you manage your network resources, consider the fictional company SiteRequest. SiteRequest recently added a new wide IP, **www.SiteRequestAsia.com**, to their Tokyo data center, and the IT department wants to see how the Link Controller system at that data center load balances traffic for that wide IP across multiple resources. By using the wide IP member statistics available through the Link Controller system, the IT department can monitor the performance of the resources that belong to the wide IP, allowing them to determine if more resources are required for the new wide IP.

The server statistics screen consists of a Wide IP Member Statistics table. This table contains the following information:

◆ Status

The Status column indicates the current status of the wide IP member. The available status types are: Available, Unavailable, Offline, and Unknown. Each status type is represented by a symbol; for example, the available status type is represented by a green circle.

◆ Wide IP Member

The Wide IP Member column displays the name or IP address of a resource that belongs to a wide IP. Each name displays as a link. When you click the link, the properties screen for the resource opens.

◆ Wide IP

The Wide IP column displays the name of the wide IP to which the resource belongs. Each wide IP displays as a link. When you click the link, the properties screen for the wide IP opens.

◆ Load Balancing

The Load Balancing column provides information on how the Link Controller system has load balanced previous connection requests to this resource. This column consists of four subcolumns:

- The Preferred subcolumn displays the cumulative number of requests that the Link Controller system load balanced with the preferred load balancing method.
- The Alternate subcolumn displays the cumulative number of requests that the Link Controller system load balanced with the alternate load balancing method.
- The Fallback subcolumn displays the cumulative number of requests that the Link Controller system load balanced with the Fallback load balancing method.

Link statistics

Link statistics focus on how much traffic is flowing in and out through a specific link to the Internet. This information can help you prevent a link from being over-used, saving your organization from higher bandwidth costs. You can access the link statistics by selecting **Links** from the **Statistics Type** list in the Statistics screen. For information on accessing the Statistics screen, see *Accessing statistics*, on page 12-1.

As an example of how the statistics for data centers can help you manage your network resources, consider the fictional company SiteRequest. SiteRequest has two links with two different Internet Service Providers (ISPs). The primary ISP is paid in advance for a specific amount of bandwidth usage. This allows SiteRequest to save money, but if the bandwidth exceeds the prepaid amount, the costs increase considerably. As a result, the IT department uses a second ISP, which has a slower connection but considerably lower costs. By using the links statistics, the IT department can ensure that links to the Internet are used as efficiently as possible.

The link statistics screen consists of a Link Statistics table. This table contains the following information:

◆ **Status**

The Status column indicates the current status of the link. The available status types are: Available, Unavailable, Offline, and Unknown. Each status type is represented by a symbol; for example, the available status type is represented by a green circle.

◆ **Link**

The Link column displays the name of a link for which the Link Controller system is responsible. Each name displays as a link. When you click the link, the properties screen for the link opens.

◆ **Throughput (bits/sec)**

The Throughput (bits/sec) column contains four subcolumns:

- The In column displays the cumulative number of bits per second received by the data center.
- The Out column displays the cumulative number of bits per second sent from the data center.
- The Total column displays the cumulative number of both incoming and outgoing bits per second for the link.
- The Over Prepaid displays the amount of traffic, in bits per second, that has exceeded the prepaid traffic allotment for the link.

In addition to viewing the link data as a table, you can also view it in a graph format. To use this format, click the **Graph** button. A graph screen opens, which shows the amount of traffic used over time. You can change the amount of time shown in the graph by selecting a value from the **Graph Interval** list, located in the Display Options area of the screen.

Paths statistics

The paths statistics captured by the Link Controller system provide information on how quickly traffic moves between a Local DNS and a resource for which the system is responsible. Information presented in the paths statistics screen includes details on round trip times (RTT), hops, and completion rates. You can access the paths statistics by selecting **Paths** from the **Statistics Type** list in the Statistics screen. For information on accessing the Statistics screen, see *Accessing statistics*, on page 12-1.

Paths statistics are primarily used when you employ a dynamic load balancing mode for a given wide IP or pool. You can use the information in the Paths statistics to get an overall sense of how responsive your wide IPs are in relation to the Local DNS servers that have been sending name resolution requests to a wide IP.

The paths statistics screen consists of a paths statistics table. This table contains the following information:

- ◆ **Local DNS Address**

The Local DNS Address column displays the IP address of each Local DNS that has sent a name resolution request for a wide IP for which the Link Controller is responsible.

- ◆ **Link**

The Link column displays the ISP link that the Link Controller used to send and receive data from the Local DNS.

- ◆ **Round Trip Time (RTT)**

The Round Trip Time (RTT) column contains two subcolumns:

- The Current subcolumn displays the current round trip time between the Local DNS and the Link Controller.
- The Average subcolumn displays the average round trip time between the Local DNS and the Link Controller.

- ◆ **Hops**

The Hops column contains two subcolumns:

- The Current subcolumn displays the current number of hops between the Local DNS and the Link Controller.
- The Average subcolumn displays the average number of hops between the Local DNS and the Link Controller.

- ◆ **Completion Rate**

The Completion Rate column contains two subcolumns:

- The Current subcolumn displays the current completion rate of transactions between the Local DNS and the Link Controller.
- The Average subcolumn displays the average completion rate of transactions between the Local DNS and the Link Controller.

Local DNS statistics

The Local DNS statistics screen provides location details related to the different Local DNS servers that communicate with the Link Controller. These statistics include the geographical location of the Local DNS as well as the last time that Local DNS accessed the Link Controller. You can access the local DNS statistics by selecting **Local DNS** from the **Statistics Type** list in the Statistics screen.

As an example of how the statistics for servers can help you manage your network resources, consider the fictional company SiteRequest. SiteRequest is currently considering whether it needs a new data center in North America to ensure that its customers can access SiteRequest's Web site as effectively as possible. To help make their decision, the IT department uses the Local DNS statistics to see where most of their European traffic is coming from. By using these statistics, the IT department discovers that a high concentration of Local DNS servers accessing SiteRequest is in the southwest United States. This information proves helpful in determining that a new data center in Las Vegas might be appropriate.

The local DNS statistics screen consists of a local DNS statistics table. This table contains the following information:

◆ **IP Address**

The IP Address column displays the IP address of each Local DNS that has sent a name resolution request for a wide IP for which the Link Controller is responsible.

◆ **Requests**

The Requests column displays the number of times this Local DNS has made a name resolution request that the Link Controller handled.

◆ **Last Accessed**

The Last Accessed column displays the last time the Local DNS attempted a connection to the Link Controller.

◆ **Location**

The Location column contains four subcolumns:

- The Continent subcolumn displays the continent on which the Local DNS resides.
- The Country subcolumn displays the country in which the Local DNS is located.
- The State subcolumn displays the state in which the Local DNS is located.
- The City subcolumn displays the city in which the Local DNS is located.



13

Understanding Profiles

- Introducing profiles
- Creating and modifying profiles
- Viewing and deleting profiles
- Implementing a profile
- For more information

Introducing profiles

The BIG-IP® Link Controller™ system can manage application-specific network traffic in a variety of ways, depending on the protocols and services being used.

For each type of traffic that you want to manage, the BIG-IP system contains configuration tools that you can use to intelligently control the behavior of that traffic. These tools are called profiles. A **profile** is a system-supplied configuration tool that enhances your capabilities for managing application-specific traffic. More specifically, a profile is an object that contains user-configurable settings, with default values, for controlling the behavior of a particular type of network traffic, such as HTTP connections. After configuring a profile, you associate the profile with a virtual server. The virtual server then processes traffic according to the values specified in the profile. Using profiles enhances your control over managing network traffic, and makes traffic-management tasks easier and more efficient.

You can associate multiple profiles with a single virtual server. For example, you can associate a TCP profile and an HTTP profile with the same virtual server.

Profile types

The BIG-IP system provides several types of profiles. While some profile types correspond to specific protocols, such as HTTP and FTP, other profiles pertain to traffic behaviors applicable to multiple protocols. Table 13.1 lists the available profile types, with descriptions.

Profile Type	Description
Services profiles	
HTTP	Defines the behavior of HTTP traffic.
FTP	Defines the behavior of FTP traffic.
DNS	Defines how the system handles DNS traffic.
Persistence profiles	
Destination Address Affinity	Implements session persistence based on the destination IP address specified in the header of a client request. Also known as sticky persistence.
Source Address Affinity	Implements session persistence based on the source IP address specified in the header of a client request. Also known as simple persistence.
Protocol profiles	
Fast L4	Defines the behavior of Layer 4 IP traffic.

Table 13.1 Available profile types on the BIG-IP system

Profile Type	Description
HTTP Class	Forwards traffic to a destination based on examining traffic headers or content, using criteria that you specify.
TCP	Defines the behavior of TCP traffic.
UDP	Defines the behavior of UDP traffic.
Other profiles	
Statistics	Provides user-defined statistical counters.

Table 13.1 Available profile types on the BIG-IP system

Default profiles

The BIG-IP system includes one or more default profiles for each profile type listed in Table 13.1. A **default profile** is a system-supplied profile that contains default values for its settings. An example of a default profile is the **http** default profile. You can use a default profile in several ways:

- **You can use a default profile as is**
You simply configure your virtual server to reference the default profile.
- **You can modify the default profile settings (not recommended)**
When you modify a default profile, you lose the original default profile settings. Thus, any custom profiles you create in the future that are based on that default profile inherit the modified settings.
- **You can create a custom profile, based on the default profile (recommended)**
This allows you to preserve the default profile, and instead configure personalized settings in the custom profile. Custom profiles inherit some of the setting values of a parent profile that you specify. After creating a custom profile, you can configure your virtual server to reference the custom profile instead of the default profile. For more information on custom profiles, see *Custom and parent profiles*, following.

◆ Note

You can modify a default profile, but you cannot create or delete a default profile.

Custom and parent profiles

A **custom profile** is a profile that is derived from a parent profile that you specify. A **parent profile** is a profile from which your custom profile inherits its settings and their default values.

When creating a custom profile, you have the option of changing one or more setting values that the profile inherited from the parent profile. In this way, you can pick and choose which setting values you would like to change and which ones you would like to retain. An advantage to creating a custom profile is that by doing so, you preserve the setting values of the parent profile.

◆ Note

*If you do not specify a parent profile when you create a custom profile, the BIG-IP system automatically assigns a related default profile as the parent profile. For example, if you create a custom HTTP type of profile, the default parent profile is the default profile **http**.*

Using the default profile as the parent profile

A typical profile that you can specify as a parent profile when you create a custom profile is a default profile. For example, if you create a custom TCP-type profile called **my_tcp_profile**, you can use the default profile **tcp** as the parent profile. In this case, the BIG-IP system automatically creates the profile **my_tcp_profile** so that it contains the same settings and default values as the default profile **tcp**. The new custom profile thus inherits its settings and values from its parent profile. You can then retain or change the inherited setting values in the custom profile to suit your needs.

Using a custom profile as the parent profile

When creating a custom profile, you can specify another custom profile, rather than the default profile, as the parent profile. The only restriction is that the custom profile that you specify as the parent must be of the same profile type as the profile you are deriving from the parent. Once you have created the new custom profile, its settings and default values are automatically inherited from the custom profile that you specified as the parent.

For example, if you create a profile called **my_tcp_profile2**, you can specify the custom profile **my_tcp_profile** as its parent. The result is that the default setting values of profile **my_tcp_profile2** are those of its parent profile **my_tcp_profile**.

If you subsequently modify the settings of the parent profile (**my_tcp_profile**), the BIG-IP system automatically propagates those changes to the new custom profile.

For example, if you create the custom profile **my_tcp_profile** and use it as a parent profile to create the custom profile **my_tcp_profile2**, any changes you make later to the parent profile **my_tcp_profile** are automatically

propagated to profile **my_tcp_profile2**. Conversely, if you modify any of the settings in the new custom profile (in our example, **my_tcp_profile2**), the new custom profile does not inherit values from the parent profile for those particular settings that you modified.

Summarizing profiles

Profiles are a configuration tool that you can use to affect the behavior of certain types of network traffic. By default, the BIG-IP system provides you with a set of profiles that you can use as is. These profiles contain various settings that define the behavior of different types of traffic. Profiles also give you a way to enable connection and session persistence. Once you have assigned a profile to a virtual server, the BIG-IP system manages any traffic that corresponds to that profile type according to the settings defined in that profile.

There are two possible types of profiles: default profiles, which the BIG-IP system supplies, and custom profiles, which you typically create.

To help you better manage HTTP and TCP traffic specifically, the BIG-IP system includes a set of F5-created custom profiles. These profiles contain recommended configurations that you would most likely want to use. By using these profiles, you do not need to create them yourself.

Default profiles are useful when the values contained in them are sufficient for your needs. Custom profiles are useful when you want your values to differ from those contained in the default profile. To ease your task of configuring and maintaining profiles, the BIG-IP system ensures that a custom profile automatically inherits settings and values from a parent profile.

When you create profiles to manage a type of network traffic, you can use them in the following ways:

- You do not need to take any action to use the default profiles that are enabled by default. The BIG-IP system uses them to automatically direct the corresponding traffic types according to the values specified in the those profiles.
- You can create a custom profile, using the default profile as the parent profile, modifying some or all of the values defined in that profile.
- You can create a custom profile to use as a parent profile for other custom profiles.

Creating and modifying profiles

As described in the previous section, profiles are a configuration tool to help you manage your application traffic. To make use of profiles, you can either use the default profiles that the BIG-IP system provides, or you can create your own custom profiles. You can also modify existing profiles as needed.

More specifically, you can:

- Use a default profile as is.
- Modify a default profile.
- Create a custom profile.
- Modify a custom profile.

The following sections contain the procedures for creating and modifying profiles. To understand individual profile settings and their effect on different types of traffic, see either the remainder of this chapter, or one of the following chapters:

- Chapter 14, *Managing HTTP and FTP Traffic*
- Chapter 16, *Managing Protocol Profiles*
- Chapter 17, *Using the Statistics Profile*

For background information on default and custom profiles, see *Introducing profiles*, on page 13-1.

Using a default profile as is

The BIG-IP system provides a default profile that you can use as is for each type of traffic. A default profile includes default values for any of the properties and settings related to managing that type of traffic. To implement a default profile, you simply assign the profile to a virtual server, using the Configuration utility. You are not required to configure the setting values. For more information, see *Implementing a profile*, on page 13-10.

For information on creating or modifying a virtual server, see Chapter 6, *Configuring Virtual Servers*.

Modifying a default profile

Using the Configuration utility, you can modify the values of a default profile. We do not recommend this. Although modifying a default profile appears to be simpler and quicker than creating a custom profile, be aware that in so doing, you lose the original values. If you want to reset the profile back to its original state, you must do this manually by modifying the settings of the default profile again to specify the original values. (To find the original default values, see the relevant profile chapter in this guide, or see the online help.)

Modifying and implementing a default profile is a two-step process:

- First, you must modify the settings of the default profile, using the Configuration utility. For more information, see *To modify a default profile*, following.
- Second, you must associate that profile with a virtual server. For information on associating a profile with a virtual server, see *Implementing a profile*, on page 13-10.

To modify a default profile

1. On the Main tab, expand **Local Traffic**, and click **Profiles**. The HTTP Profiles List screen opens.
2. Select the default profile that you want to modify:
 - If you are modifying the **http** profile, click the name **http**. The properties and settings of the default **http** profile display.
 - If you are modifying a default profile other than the **http** profile, click the appropriate profile menu on the menu bar and choose a profile type. Then click a profile name. The properties and settings of that default profile display.
3. Modify the settings to suit your needs.
4. Click **Update**.

Creating a custom profile

If you do not want to use a default profile as is or change its settings, you can create a custom profile. Creating a custom profile and associating it with a virtual server allows you to implement your own specific set of traffic-management policies.

When you create a custom profile, the profile is a child profile and automatically inherits the setting values of a parent profile that you specify. However, you can change any of the values in the child profile to better suit your needs. For background information on custom profiles and inheritance of setting values, see *Custom and parent profiles*, on page 13-3.

If you do not specify a parent profile, the BIG-IP system uses the default profile that matches the type of profile you are creating.

Implementing a custom profile is a two-step process:

- First, you must create the custom profile, using the Configuration utility. For more information, see *To create a custom profile*, following.

- Second, you must associate that profile with a virtual server. For information on associating a profile with a virtual server, see *Implementing a profile*, on page 13-10.

◆ **Important**

Within the Configuration utility, each profile creation screen contains a check box to the right of each profile setting. When you check a box for a setting and then specify a value for that setting, the profile then retains that value, even if you change the corresponding value in the parent profile later. Thus, checking the box for a setting ensures that the parent profile never overwrites that value through inheritance.

To create a custom profile

1. On the Main tab, expand **Local Traffic**, and click **Profiles**.
The HTTP Profiles List screen opens.
2. Select the type of profile you want to create:
 - If you are creating an HTTP type of profile, proceed to Step 3.
 - If you are creating another type of profile, click a profile category on the menu bar and choose a profile type.
3. On the right side of the screen, click **Create**.
4. In the **Name** box, type a unique name for your profile.
5. From the **Parent Profile** list, select a profile from the list.
You can select either the default profile or another custom profile.
6. Specify, modify, or retain values for all settings:
 - If you want to specify or modify a value, locate the setting, click the box in the Custom column on the right side of the screen, and then type or modify a value.
 - If you want to retain a value inherited from the parent profile, leave the setting as is. Do not check the box in the Custom column.
7. Click **Finished**.

◆ **Tip**

*An alternative way to access the New Profile screen in the Configuration utility is to locate the Main tab, expand **Local Traffic**, click the **Create** button adjacent to the **Profiles** menu item, and select a profile type.*

Modifying a custom profile

Once you have created a custom profile, you can use the Configuration utility to adjust the settings of your custom profile later if necessary. If you have already associated the profile with a virtual server, you do not need to perform that task again.

Important

Within the Configuration utility, each profile creation screen contains a check box to the right of each profile setting. When you check a box for a setting and then specify a value for that setting, the profile then retains that value, even if you change the corresponding value in the parent profile later. Thus, checking the box for a setting ensures that the parent profile never overwrites that value through inheritance.

To modify custom profile settings

1. On the Main tab, expand **Local Traffic**, and click **Profiles**.
The HTTP Profiles List screen opens.
2. Use the menu bar to select the type of profile you want to modify.
Either click **Persistence**, or choose a profile type from either the Services or Protocols menu.
3. In the Name column, click the name of the profile you want to modify.
The settings and values for the profile display.
4. Modify or retain values for all settings:
 - If you want to modify a value, locate the setting, click the box in the Custom column on the right side of the screen, and then modify the value.
 - If you want to retain a value inherited from the parent profile, leave the setting as is. Do not check the box in the Custom column.
 - If you want to reset a value back to the parent profile value, clear the check box in the Custom column on the right side of the screen.
5. Click the **Update** button.

Viewing and deleting profiles

You can use the Configuration utility to view a list of profiles or delete a profile from the system.

Viewing a list of profiles

You can view a list of existing profiles. When you display a list of profiles, the Configuration utility displays the following information about each profile:

- Profile name
- Type of profile (persistence profiles only)
- Parent profile

Use the following procedure to view a list of profiles defined on the BIG-IP system.

◆ **Tip**

*When listing existing profiles, you can use the **Search** box that displays directly above the profile list. With the **Search** box, you can specify a string to filter the list, thereby showing only those objects that match the string. The default setting is an asterisk (*), which means show all objects.*

To view a list of profiles

1. On the Main tab, expand **Local Traffic**, and click **Profiles**.
The HTTP Profiles List screen opens.
2. Use the menu bar to select the category of profile you want to view.
For example, click **Persistence** to view a list of Persistence profiles.
Or, to view a list of TCP profiles, from the Protocol menu, choose TCP.
The list screen opens for that profile type.

Deleting a profile

You can delete any existing profile except a default profile.

To delete a profile

1. Display the pertinent list of profiles, using the previous procedure.
2. Click the Select box to the left of the custom profile that you want to delete.
3. Click **Delete**.
The Delete confirmation screen opens.
4. Click **Delete**.

Implementing a profile

Once you have created a profile for a specific type of traffic, you implement the profile by associating that profile with one or more virtual servers.

You associate a profile with a virtual server by configuring the virtual server to reference the profile. Whenever the virtual server receives that type of traffic, the BIG-IP system applies the profile settings to that traffic, thereby controlling its behavior. Thus, profiles not only define capabilities per network traffic type, but also ensure that those capabilities are available for a virtual server.

To assign a profile to a virtual server

1. On the Main tab, expand **Local Traffic**, and click **Virtual Servers**.
The Virtual Server List screen opens.
2. In the Name column, click a virtual server name.
The properties and settings of the virtual server display.
3. Locate the setting for the type of profile you want to assign and select the name of a default or custom profile.
4. At the bottom of the screen, click **Update**.

◆ Note

You can also assign a profile to a virtual server at the time that you create the virtual server.

Because certain kinds of traffic use multiple protocols and services, users often create multiple profiles and associate them with a single virtual server.

For example, a client application might use the TCP and HTTP protocols and services to send a request. This type of traffic would therefore require three profiles, based on the three profile types TCP and HTTP.

Each virtual server lists the names of the profiles currently associated with that virtual server. You can add or remove profiles from the profile list, using the Configuration utility.

The BIG-IP system has specific requirements regarding the combinations of profile types allowed for a given virtual server. Table 13.2 shows the specific combinations of profile types that you can configure on a virtual server.

Profile Type	Prerequisite Profiles	Incompatible Profiles
Protocol profiles		
Fast L4	None	All
TCP	None	UDP, Fast L4, Fast L7
UDP	None	TCP, Fast L4, Fast L7
Services profiles		
HTTP	TCP	FTP
FTP	TCP	HTTP
Persistence profiles		
Destination Address Affinity	Any	None
Source Address Affinity	Any	None
Other profiles		
Statistics	TCP	N/A

Table 13.2 Profile combinations that the BIG-IP system allows and disallows

In directing traffic, if a virtual server requires a specific type of profile that does not appear in its profile list, the BIG-IP system uses the relevant default profile, automatically adding the profile to the profile list. For example, if a client application sends traffic over TCP and HTTP, and you have assigned only HTTP profiles, the BIG-IP system automatically adds the default profile **tcp** to its profile list.

At a minimum, a virtual server must reference a profile, and that profile must be associated with a UDP, FastL4, or TCP profile type. Thus, if you have not associated a profile with the virtual server, the BIG-IP system adds a UDP, FastL4, or TCP default profile to the profile list.

The default profile that the BIG-IP system chooses depends on the configuration of the virtual server's protocol setting. If the protocol setting is set to **UDP**, the BIG-IP system adds the **udp** profile to its profile list. If the protocol setting is set to anything other than **UDP**, the BIG-IP system adds the **FastL4** profile to its profile list.

For more information

For information on configuring specific types of profiles, see the following chapters:

- Chapter 14, *Managing HTTP and FTP Traffic*
- Chapter 15, *Enabling Session Persistence*
- Chapter 16, *Managing Protocol Profiles*
- Chapter 17, *Using the Statistics Profile*



14

Managing HTTP and FTP Traffic

- Introducing HTTP and FTP traffic management
- Configuring HTTP standard profile settings
- Configuring FTP profile settings

Introducing HTTP and FTP traffic management

The BIG-IP® Local Traffic Manager™ system that includes the Link Controller™ offers several features that you can use to intelligently control your HTTP, HTTPS, and FTP traffic. Examples of these features are the insertion of headers into HTTP requests and the compression of HTTP server responses.

These features are available through configuration of HTTP or FTP profiles. A **profile** is a group of settings, with values, that corresponds to a specific type of traffic, such as HTTP traffic. A profile defines the way that you want the BIG-IP system to manage that traffic type.

In addition to HTTP and FTP profiles, the BIG-IP system includes other features to help you manage your application traffic, such as health monitors for checking the health of HTTP and FTP services, and iRules® for querying or manipulating HTTP header or content data.

Table 14.1 summarizes the capabilities within the BIG-IP system for managing HTTP and FTP traffic and shows the BIG-IP object that you configure to implement each feature.

Feature	Description	Configuration Object
Monitoring of HTTP, HTTPS, and FTP ports on pool members	You can associate an HTTP, HTTPS, or FTP monitor with the members of a pool to ensure that pool members are ready and able to receive traffic on specific ports.	Load balancing pool HTTP, HTTPS, and FTP health monitors
Session persistence	You can ensure that HTTP sessions persist to the same pool member across connections.	Persistence profile and iRule
Redirection of HTTP requests	By configuring an HTTP profile or writing an iRule, you can instruct the BIG-IP system to redirect HTTP traffic based on URI, status code, or content.	HTTP Profile and iRule
Chunking of requests and responses	You can configure the BIG-IP system to unchunk or rechunk HTTP responses.	HTTP profile
Pipelining	The BIG-IP system supports HTTP pipelining.	HTTP profile
IPv4-to-IPv6 compatibility	Ensures compatibility between IP version 4 and IP version 6 clients and servers when using the FTP protocol.	FTP profile

Table 14.1 Summary of the BIG-IP system features related to HTTP and FTP traffic control

Configuring HTTP standard profile settings

You can configure HTTP profiles to ensure that HTTP traffic management suits your specific needs. These configuration settings are organized into several categories on the New HTTP Profile screen in the Configuration utility: General Properties, Settings, Compression, and RAM Cache. You can configure these settings when you create a profile, or after profile creation by modifying the profile's settings. For specific procedures on configuring a profile, see Chapter 13, *Understanding Profiles*.

For the profile settings that appear in the General Properties and Settings areas of the HTTP Profile screen, you can specify values where none exist, or modify any default values to suit your needs. For information about other HTTP profile settings, see *Configuring FTP profile settings*, on page 14-6.

Understanding HTTP profile settings

You can use the default **http** profile as is, or create a custom HTTP profile. The **http** profile is considered a default profile because it does not inherit setting values from a parent profile.

Table 14.2 shows the profile settings for an HTTP type of profile. For those settings that have default values, you can retain those default settings or modify them. Following this table are descriptions of the settings and the procedure for changing them.

Setting	Description	Default Value
Name	Specifies the user-supplied name of the profile. You must specify a name for your profile.	No default value
Parent Profile	Specifies the profile from which your custom profile is derived.	http
Request Chunking	Specifies how to handle chunking for HTTP requests. Possible values are Rechunk , Selective , and Preserve .	Preserve
Response Chunking	Specifies how to handle chunking for HTTP responses. Possible values are Unchunk , Rechunk , Selective , and Preserve .	Selective
Redirect Rewrite	Allows you to modify HTTP redirections. Possible values are Matching , All , Nodes , or None .	None
Encrypt Cookies	Specifies the cookie names for the system to encrypt.	No default value
Cookie Encryption Passphrase	Specifies a passphrase for cookie encryption.	No default value
Confirm Cooke Encryption Passphrase	Specifies the passphrase for cookie encryption again.	No default value

Table 14.2 *Settings of an HTTP profile*

Setting	Description	Default Value
Maximum Header Size	Specifies the maximum size in bytes that the BIG-IP system allows for HTTP headers.	32768
Maximum Header Count	Specifies the maximum number of headers in an HTTP request or response that the system handles. If a request or response contains more headers than the number specified, the system drops the connection.	64
Pipelining	Enables or disables HTTP pipelining.	Enabled
Insert XForwarded-For	Inserts an XForwarded-For header into an HTTP request, to use with connection pooling. This feature adds the IP address of the client as the value of the XForwarded-For header.	Disabled
LWS Maximum Columns	Specifies the maximum width allowed for an HTTP header that is inserted into an HTTP request.	80
LWS Separator	Specifies the separator that the BIG-IP system should use between HTTP headers when a header exceeds the maximum width allowed.	\r\n
Maximum Requests	Specifies the maximum number of HTTP requests that the system allows for a single Keep-Alive connection.	0
Send Proxy Via Header In Request	Specifies whether to Remove , Preserve , or Append Via headers included in a client request to an origin web server.	Remove
Send Proxy Via Header In Response	Specifies whether to Remove , Preserve , or Append Via headers included in an origin web server response to a client.	Remove

Table 14.2 *Settings of an HTTP profile*

Before configuring an HTTP profile, it is helpful to have a description of certain settings that you might want to change.

Specifying a profile name

To create an HTTP profile, you must specify a unique name for the profile. The **Name** setting is one of only two settings for which you must actively specify a value when creating an HTTP profile; all other settings have default values.

To specify a profile name, simply locate the **Name** setting and type a unique name for the profile.

Specifying a parent profile

Every profile that you create is derived from a parent profile. You can use the default **http** profile as the parent profile, or you can use another HTTP profile that you have already created.

To specify a parent profile, locate the **Parent Profile** setting and select a profile name.

Configuring chunking

Sometimes, you might want to inspect and/or modify HTTP application data, such as compressing the content of an HTTP response. Such inspections or modifications require that the response be *unchunked*, that is, not in chunked encoding. Using the **Response Chunking** settings, the BIG-IP system can unchunk a chunked response before performing an action on that response.

Possible values for this setting are **Unchunk**, **Rechunk**, **Selective**, and **Preserve**. The default value is **Selective**.

Table 14.3 describes each of these values and the action that the BIG-IP system takes, depending on whether an original response is chunked or unchunked.

Setting	Original response is chunked	Original response is unchunked
Unchunk	The BIG-IP system unchunks the response and processes the HTTP content, and passes the response on as unchunked. The connection closes when all data is sent to the client as indicated by the Connection: Close header.	The BIG-IP system processes the HTTP content and passes the response on untouched.
Rechunk	The BIG-IP system unchunks the response, processes the HTTP content, re-adds the chunk trailer headers, and then passes the response on as chunked. Any chunk extensions are lost.	The BIG-IP system adds transfer encoding and chunking headers on egress.
Selective	Same as Rechunk .	The BIG-IP system processes the HTTP content and then passes the response on untouched.
Preserve	The BIG-IP system leaves the response chunked, processes the HTTP content, and passes the response on untouched. Note that if HTTP compression is enabled, the BIG-IP system does not compress the response.	The BIG-IP system processes the HTTP content and then passes the response on untouched.

Table 14.3 *Chunking behavior of the BIG-IP system*

Specifying the maximum header size

With the **Maximum Header Size** setting, you can specify the maximum size that the BIG-IP system allows for HTTP headers. The default value is **32768** and is represented in bytes.

Enabling support for pipelining

Normally, a client cannot make a request until the previous request has received a response. HTTP/1.1 pipelining allows clients to make requests even when prior requests have not received a response. For this to succeed, however, destination servers must include support for pipelining. This feature enables that support on the BIG-IP system.

To enable pipelining, locate the **Pipelining** setting and check the box. By default, this feature is set to **Enabled**.

Inserting an XForwarded For header

When using connection pooling, which allows clients to make use of existing server-side connections, you can insert the **XForwarded For** header into a request. When you configure the BIG-IP system to insert this header, the target server can identify the request as coming from a client other than the client that initiated the connection. The default setting is **Disabled**.

Configuring the maximum columns for linear white space

The **LWS Maximum Columns** setting specifies the maximum number of columns allowed for a header that is inserted into an HTTP request.

To configure the **LWS Maximum Columns** setting, specify a maximum value. The default value for this setting is **80**.

Configuring a linear white space separator

The **LWS Separator** setting specifies the separator that the BIG-IP system should use between HTTP headers when a header exceeds the maximum width specified by the **LWS Maximum Columns** setting.

To configure the **LWS Separator** setting, specify a value for the separator. This setting has no default value.

Specifying a maximum number of requests

The **Maximum Requests** setting specifies the maximum number of requests that the system allows for a single Keep-Alive connection. When the specified limit is reached, the final response contains a **Connection: close** header is followed by the closing of the connection. The default setting is **0**, which in this case means that the system allows an infinite number of requests per **Keep-Alive** connection.

Configuring FTP profile settings

You can tailor FTP profile settings to your specific needs. For those settings that have default values, you can retain those default settings or modify them. You can modify any settings either when you create the profile, or at any time after you have created it. For specific procedures on configuring a profile, see Chapter 13, *Understanding Profiles*.

Table 14.4 lists these configurable settings, along with a short description of each and the default values. Following this table are descriptions of specific settings.

General property	Description	Default Value
Name	Specifies the user-supplied name of the profile. Specifying a name for your profile is required.	No default value
Parent Profile	Specifies the profile from which your custom profile is derived.	ftp
Translate Extended	Ensures compatibility between IP version 4 and IP version 6 clients and servers when using the FTP protocol.	Enabled
Data Port	Allows the FTP service to run on an alternate port.	20

Table 14.4 Configuration settings of an FTP profile

Before configuring an FTP profile, it is helpful to have a description of certain node settings that you might want to change.

Specifying a profile name

To create an FTP profile, you must specify a unique name for the profile. The **Name** setting is one of only two settings for which you must actively specify a value when creating an FTP profile; all other settings have default values.

Specifying a parent profile

Every profile that you create is derived from a parent profile. In the **Parent Profile** setting, you can select the default **ftp** profile as the parent profile, or you can select another FTP profile that you have already created.

Specifying a Translate Extended value

Because IP version 6 addresses are not limited to 32 bits (unlike IP version 4 addresses), compatibility issues can arise when using FTP in mixed IP-version configurations.

Enabled by default, the **Translate Extended** setting causes the BIG-IP system to automatically translate FTP commands when a client-server configuration contains both IP version 4 and IP version 6 systems. For example, if a client system running IP version 4 sends the FTP **PASV** command to a server running IP version 6, the BIG-IP system automatically translates the **PASV** command to the equivalent FTP command for IP version 6 systems, **EPSV**.

The BIG-IP system translates the FTP commands **EPRV** and **PORT** in the same way.

It is highly unlikely that you need to change the default value (**Enabled**) for this setting. The only case where you might want to disable this setting is when sending an **EPSV** command to an IP version 4 system, such as when testing an FTP server.

Specifying a data port

The **Data Port** setting allows the FTP service to run on an alternate port. You can use the default port number **20**, or specify another port number.



15

Enabling Session Persistence

- Introducing session persistence
- Persistence types and their profiles

Introducing session persistence

Using a BIG-IP® Local Traffic Manager™ system that includes the Link Controller™, you can configure session persistence. When you configure ***session persistence***, the BIG-IP system tracks and stores session data, such as the specific pool member that serviced a client request. The primary reason for tracking and storing session data is to ensure that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions.

In addition, session persistence can track and store other types of information, such as user preferences or a user name and password.

The BIG-IP system offers two types of session persistence, each one designed to accommodate a specific type of storage requirement for session data. The type of persistence that you implement depends on where and how you want to store client-specific information, such as items in a shopping cart or airline ticket reservations.

For example, you might store airline ticket reservation information in a back-end database that all servers can access, or on the specific server to which the client originally connected. When you enable persistence, returning clients can bypass load balancing and instead connect to the server to which they last connected in order to access their saved information.

The BIG-IP system keeps session data for a period of time that you specify.

The primary tool for configuring session persistence is to configure a persistence profile and assign it to a virtual server. If you want to enable persistence for specific types of traffic only, as opposed to all traffic passing through the virtual server, you can write an iRule.

Configuring a persistence profile

A ***persistence profile*** is a pre-configured object that automatically enables persistence when you assign the profile to a virtual server. By using a persistence profile, you avoid having to write a program to implement a type of persistence.

Each type of persistence that the BIG-IP system offers includes a corresponding default persistence profile. These persistence profiles each contain settings and setting values that define the behavior of the BIG-IP system for that type of persistence. You can either use the default profile or create a custom profile based on the default.

For more information, see the following chapters of this guide:

- To configure persistence profiles, see *Persistence types and their profiles*, on page 15-2.
- To understand profiles in general, see Chapter 13, *Understanding Profiles*.

Enabling session persistence through iRules

Instead of configuring a persistence profile, which enables a persistence type for all sessions passing through the virtual server, you can write an iRule, which enables a persistence type for particular requests (for example, for HTTP traffic that includes a certain cookie version only).

The remainder of this chapter focuses on enabling persistence using persistence profiles. For information on enabling persistence by writing an iRule, see the F5 Networks DevCentral web site <http://devcentral.f5.com>, and Chapter 18, *Writing iRules*.

Persistence types and their profiles

You can configure persistence profile settings to set up session persistence on the BIG-IP system. You can configure these settings when you create a profile or after profile creation by modifying the profile's settings. For specific procedures on configuring a profile, see Chapter 13, *Understanding Profiles*.

Types of persistence

The persistence types that you can enable using a persistence profile are:

- ◆ **Destination address affinity persistence**
Also known as sticky persistence, destination address affinity persistence supports TCP and UDP protocols, and directs session requests to the same server based solely on the destination IP address of a packet.
- ◆ **Source address affinity persistence**
Also known as simple persistence, source address affinity persistence supports TCP and UDP protocols, and directs session requests to the same server based solely on the source IP address of a packet.

Understanding criteria for session persistence

Regardless of the type of persistence you are implementing, you can specify the criteria that the BIG-IP system uses to send all requests from a given client to the same pool member. These criteria are based on the virtual server or servers that are hosting the client connection. To specify these criteria, you use the **Match Across Services**, **Match Across Virtual Servers**, and **Match Across Pools** profile settings. Before configuring a persistence profile, it is helpful to understand these settings.

Specifying the Match Across Services setting

When you enable the **Match Across Services** profile setting, the BIG-IP system attempts to send all persistent connection requests received from the same client, within the persistence time limit, to the same pool member only when the virtual server hosting the connection has the same virtual address as the virtual server hosting the initial persistent connection. Connection requests from the client that go to other virtual servers with different virtual addresses, or those connection requests that do not use persistence, are load balanced according to the load balancing method defined for the pool.

For example, suppose you configure virtual server mappings where the virtual server **v1:http** has persistence enabled and references the **http_pool** (containing the nodes **n1:http** and **n2:http**) and the virtual server **v1:ssl** has persistence enabled and references the pool **ssl_pool** (containing the nodes **n1:ssl** and **n2:ssl**).

Suppose the client makes an initial connection to **v1:http**, and the load balancing algorithm assigned to the pool **http_pool** chooses **n1:http** as the node. If the client subsequently connects to **v1:ssl**, the BIG-IP system uses the persistence session established with the first connection to determine the pool member that should receive the connection request, rather than the load balancing method. The BIG-IP system should then send the third connection request to **n1:ssl**, which uses the same node as the **n1:http** node that currently hosts the client's first connection with which it shares a persistent session.

If the same client then connects to a virtual server with a different virtual address (for example, **v2:ssl**), the BIG-IP system starts tracking a new persistence session, using the load balancing method to determine which node should receive the connection request. The system starts a new persistence session because the requested virtual server uses a different virtual address (**v2**) than the virtual server hosting the first persistent connection request (**v1**).

Important

*In order for the **Match Across Services** setting to be effective, virtual servers that use the same virtual address, as well as those that use TCP persistence, should include the same node addresses in the virtual server mappings.*

Specifying the Match Across Virtual Servers setting

You can set the BIG-IP system to maintain persistence for all sessions requested by the same client, regardless of which virtual server hosts each individual connection initiated by the client. When you enable the **Match Across Virtual Servers** setting, the BIG-IP system attempts to send all persistent connection requests received from the same client, within the

persistence time limit, to the same pool member. Connection requests from the client that do not use persistence are load balanced according to the currently selected load balancing method.

Important

In order for this setting to be effective, virtual servers that use pools with TCP persistence should include the same member addresses in the virtual server mappings.

Specifying the Match Across Pools setting

When you enable the **Match Across Pools** setting, the BIG-IP system can use any pool that contains a given persistence record. The default is disabled (cleared).

Destination address affinity persistence

You can optimize your server array with destination address affinity persistence. **Destination address affinity persistence**, also known as sticky persistence, directs requests for a certain destination IP address to the same server, regardless of which client made the request.

This type of persistence provides the most benefits when load balancing caching servers. A caching server intercepts web requests and returns a cached web page if it is available. In order to improve the efficiency of the cache on these servers, it is necessary to send similar requests to the same server repeatedly. You can use the destination address affinity persistence type to cache a given web page on one server instead of on every server in an array. This saves the other servers from having to duplicate the web page in their cache, wasting memory.

Understanding Destination Address Affinity profile settings

To implement destination address affinity persistence, you either use the default **dest_addr** profile or create a custom profile. Table 15.1 shows the settings and their values that make up a Destination Address Affinity profile.

Setting	Description	Default Value
Name	Specifies a unique name for the profile. This setting is required.	No default value
Persistence Type	Specifies the type of persistence profile. This setting is required.	Destination Address Affinity
Parent Profile	Specifies the existing profile that you want to use as the source. The new profile inherits all of the settings and values from the specified parent profile, except ones you specifically change on the properties screen.	dest_addr

Table 15.1 Settings of a Destination Address Affinity persistence profile

Setting	Description	Default Value
Match Across Services	Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.	Disabled (Cleared)
Match Across Virtual Servers	Specifies that all persistent connections from the same client IP address go to the same node.	Disabled (Cleared)
Match Across Pools	Specifies that the BIG-IP system can use any pool that contains this persistence entry.	Disabled (Cleared)
Hash Algorithm	Specifies the algorithm the system uses for hash persistence load balancing. The hash result is the input for the algorithm. Possible values are: Default: Specifies that the system uses the index of pool members to obtain the hash result for the input to the algorithm. CARP: Specifies that the system uses the Cache Array Routing Protocol (CARP) to obtain the hash result for the input to the algorithm.	default
Mask	Specifies the mask that the BIG-IP system should use before matching with an existing persistence entry.	255.255.255.255
Timeout	Specifies the duration of the persistence entries. Possible values are: Specify: Specifies the number of seconds before the persistence entry times out. Indefinite: Specifies that the persistence entry does not time out.	180
Override Connection Limit	Specifies whether pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.	Disabled (Cleared)

Table 15.1 Settings of a Destination Address Affinity persistence profile

Source address affinity persistence

Source address affinity persistence, also known as simple persistence, tracks sessions based only on the source IP address. When a client requests a connection to a virtual server that supports source address affinity persistence, the BIG-IP system checks to see if that client previously connected, and if so, returns the client to the same pool member.

Persistence settings apply to all protocols. When the persistence timer is set to a value greater than **0**, persistence is **on**. When the persistence timer is set to **0**, persistence is **off**.

The persistence mask feature works only for virtual servers that implement source address affinity persistence. By adding a persistence mask, you identify a range of source IP addresses to manage together as a single source address affinity persistent connection when connecting to the pool.

Understanding Source Address Affinity persistence profile settings

To implement source address affinity persistence, you can either use the default **source_addr** profile or create a custom profile. Table 15.2 shows the settings and values that make up a Source Address Affinity profile.

Setting	Description	Default Value
Name	Specifies a unique name for the profile. This setting is required.	No default value
Persistence Type	Specifies the type of persistence profile. This setting is required.	Source Address Affinity
Parent Profile	Specifies the existing profile that you want to use as the source. The new profile inherits all of the settings and values from the specified parent profile, except ones you specifically change on the properties screen.	source_addr
Match Across Services	Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.	Disabled (Cleared)
Match Across Virtual Servers	Specifies that all persistent connections from the same client IP address go to the same node.	Disabled (Cleared)
Match Across Pools	Specifies that the BIG-IP system can use any pool that contains this persistence entry.	Disabled (Cleared)
Hash Algorithm	Specifies the algorithm the system uses for hash persistence load balancing. The hash result is the input for the algorithm. Possible values are: Default: Specifies that the system uses the index of pool members to obtain the hash result for the input to the algorithm. CARP: Specifies that the system uses the Cache Array Routing Protocol (CARP) to obtain the hash result for the input to the algorithm.	default
Timeout	Specifies the duration of the persistence entries. Possible values are: Specify: Specifies the number of seconds before the persistence entry times out. Indefinite: Specifies that the persistence entry does not time out.	180
Mask	Specifies the mask that the BIG-IP system should use before matching with an existing persistence entry.	None
Map Proxies	Enables or disables proxy mapping.	Enabled (Checked)
Override Connection Limit	Specifies whether pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.	Disabled (Cleared)

Table 15.2 Settings of a Source Address Affinity persistence profile



16

Managing Protocol Profiles

- Introducing protocol profiles
- Configuring a Fast L4 profile
- Configuring a TCP profile
- Configuring a UDP profile

Introducing protocol profiles

Some of the profiles that you can configure are known as Protocol profiles. The Protocol profiles types are:

- Fast L4
- TCP
- UDP

For each Protocol profile type, the BIG-IP® Link Controller™ system provides a pre-configured profile with default settings. In most cases, you can use these default profiles as is. If you want to change these settings, you can configure protocol profile settings when you create a profile, or after profile creation by modifying the profile's settings.

The remainder of this chapter lists the traffic-management settings contained in the Fast L4, TCP, and UDP profiles. For information on configuring other types of profiles, see the following:

- Chapter 14, *Managing HTTP and FTP Traffic*.
- Chapter 15, *Enabling Session Persistence*.
- Chapter 17, *Using the Statistics Profile*.

Configuring a Fast L4 profile

The purpose of a Fast L4 profile is to help you manage Layer 4 traffic more efficiently. When you assign a Fast L4 profile to a virtual server, the Packet Velocity ASIC® (PVA) hardware acceleration within the BIG-IP system can process some or all of the Layer 4 traffic passing through the system. By offloading Layer 4 processing to the PVA hardware acceleration, the BIG-IP system can increase performance and throughput for basic routing functions (Layer 4) and application switching (Layer 7).

You can use a Fast L4 profile with these types of virtual servers-- Performance (Layer 4), Forwarding (Layer 2), and Forwarding (IP). Therefore, you can use a Fast L4 profile when you do *not* need the following traffic management features:

- HTTP optimizations
- TCP optimizations
- iRules® for non-Layer 4 events
- HTTP pipelining

Understanding Fast L4 profile settings

You can use the default **fastl4** profile as is, or create a custom Fast L4 profile. For your typical needs, most of the default values for the Fast L4 profile settings suffice. The specific settings that you might want to change are **Reset on Timeout** and **Idle Timeout**.

◆ Note

*Any changes you make to an existing Fast L4 profile take effect on a connection only after the **Idle Timeout** value has expired or the connection is closed.*

Table 16.1 lists and describes the settings of a Fast L4 profile.

Setting	Description	Default Value
Name	This setting specifies a unique name for the profile.	No default value
Parent Profile	This setting specifies the profile that you want to use as the parent profile. Your new profile inherits all non-custom settings and values from the parent profile specified.	fastL4
Reset on Timeout	If this setting is enabled and a TCP connection exceeds the timeout value for idle connections, the BIG-IP system sends a reset in addition to deleting the connection.	Enabled
Reassemble IP Fragments	If this setting is enabled, the BIG-IP system reassembles IP fragments.	Disabled

Table 16.1 Settings of a Fast L4 profile

Setting	Description	Default Value
Idle Timeout	This setting specifies the number of seconds that a connection is idle before the connection is eligible for deletion.	300
TCP Handshake Timeout	<p>Specify: Specifies the acceptable duration for a TCP handshake, that is, the maximum idle time between a client SYN and a client ACK. If the TCP handshake takes longer than the timeout, the system automatically closes the connection.</p> <p>Disabled: Specifies that the system does not apply a timeout to a TCP handshake.</p> <p>Indefinite: Specifies that the acceptable duration for a TCP handshake is indefinite.</p>	5
Max Segment Size Override	<p>Overrides the maximum segment size (MSS), which is 1460. Possible values are:</p> <p>Disabled: Specifies that you want the maximum segment size to remain at 1460.</p> <p>Specify. Permits you to override the maximum segment size (1460) by specifying a number. Note that specifying a 0 value is equivalent to retaining the default value (Disabled).</p>	Disabled
IP ToS to Client	This setting specifies the Type of Service level that the BIG-IP system assigns to UDP packets when sending them to clients.	Pass Through
IP ToS to Server	This setting specifies the Type of Service level that the BIG-IP system assigns to UDP packets when sending them to servers	Pass Through
Link QoS to Client	This setting specifies the Quality of Service level that the BIG-IP system assigns to UDP packets when sending them to clients.	Pass Through
Link QoS to Server	This setting specifies the Quality of Service level that the BIG-IP system assigns to UDP packets when sending them to servers.	Pass Through
TCP Timestamp Mode	Specifies the action that the BIG-IP system should take on TCP timestamps. Possible values are: Preserve , Strip , and Rewrite .	Preserve
TCP Window Scale Mode	Specifies the action that the BIG-IP system should take on TCP windows. Possible values are: Preserve , Strip , and Rewrite .	Preserve
Generate Internal Sequence Numbers	Enables the BIG-IP system to generate its own sequence numbers for SYN packets, according to RFC 1948. When enabled, this setting allows timestamp recycling.	Disabled
Strip Sack OK	Enables the BIG-IP system to block a TCP SackOK option from passing to the server on an initiating SYN.	Disabled
RTT from Client	Specifies that the BIG-IP system should use TCP timestamp options to measure the round-trip time to the client.	Disabled

Table 16.1 Settings of a Fast L4 profile

Setting	Description	Default Value
RTT from Server	Specifies that the BIG-IP system should use TCP timestamp options to measure the round-trip time to the server.	Disabled
Loose Initiation	Specifies, when checked (enabled), that the system initializes a connection when it receives any TCP packet, rather than requiring a SYN packet for connection initiation. The default is disabled. We recommend that if you enable the Loose Initiation setting, you also enable the Loose Close setting. <i>Important: Enabling loose initiation can permit stray packets to pass through the system. This can pose a security risk and reduce system performance.</i>	Disabled
Loose Close	Specifies, when checked (enabled), that the system closes a loosely-initiated connection when the system receives the first FIN packet from either the client or the server.	Disabled
TCP Close Timeout	Specifies the length of time in seconds that a connection can remain idle before deletion, once the system receives a CLOSE packet for that connection. The TCP Close Timeout value must be less than the Idle Timeout value. Also, the TCP Close Timeout value is valid only if you enable the Loose Initiation or the Loose Close settings.	5

Table 16.1 Settings of a Fast L4 profile

Configuring a TCP profile

A TCP profile is a configuration tool that helps you to manage TCP network traffic. Many of the configuration settings of a TCP profile are standard SYSCTL types of settings, while others are unique to the BIG-IP system. You can implement this profile as is, or you can change the value of the settings to suit your needs.

Understanding TCP profile settings

You can use the default **tcp** profile as is, or create a custom TCP profile. Table 16.2 lists and describes the settings of a TCP profile.

Setting	Description	Default Value
Name	Specifies a unique name for the profile.	No default value
Parent Profile	Specifies the profile that you want to use as the parent profile. Your new profile inherits all non-custom settings and values from the parent profile specified.	tcp
Reset on Timeout	If this setting is enabled and a TCP connection exceeds the timeout value for idle connections, sends a reset in addition to deleting the connection.	Enabled (Checked)
Time Wait Recycle	Recycles the connection when a SYN packet is received in a TIME-WAIT state.	Enabled (Checked)
Delayed ACKs	If this setting is enabled, allows coalescing of multiple acknowledgement (ACK) responses.	Enabled (Checked)
Proxy Maximum Segment	Advertises the same maximum segment to the server as was negotiated with the client.	Disabled (Cleared)
Proxy Options	Advertises an option (such as timestamps) to the server only if it was negotiated with the client.	Disabled (Cleared)
Proxy Buffer Low	Specifies the proxy buffer level at which the receive window was opened.	4096
Proxy Buffer High	Specifies the proxy buffer level at which the receive window was closed.	16384
Idle Timeout	Specifies the number of seconds that a connection is idle before the connection is eligible for deletion.	300
Zero Window Timeout	Specifies the timeout in milliseconds for terminating a connection with an effective zero length TCP transmit window. The timeout starts when the peer advertises a zero length TCP window or when enough data has been sent to fill the previously advertised window. The timer is canceled when a non-zero length window is received. The default is 20000 milliseconds.	20000
Time Wait	Specifies the number of milliseconds that a connection is in a TIME-WAIT state before entering the CLOSED state.	2000
FIN Wait	Specifies the number of seconds that a connection is in the FIN-WAIT or CLOSING state before quitting. A value of 0 represents a term of forever (or until the metrics of the FIN state).	5
Close Wait	Specifies the number of seconds that a connection remains in a LAST-ACK state before quitting. A value of 0 represents a term of forever (or until the metrics of the FIN state).	5
Send Buffer	Causes the BIG-IP system to send the buffer size, in bytes.	32768

Table 16.2 *Settings of a TCP profile*

Setting	Description	Default Value
Receive Window	Causes the BIG-IP system to receive the window size, in bytes.	32768
Keep Alive Interval	Causes the BIG-IP system to keep alive the probe interval, in milliseconds.	1800
Maximum SYN Retransmissions	Specifies the maximum number of retransmissions of SYN segments that the BIG-IP system allows.	3
Maximum Segment Retransmissions	Specifies the maximum number of retransmissions of data segments that the BIG-IP system allows.	8
IP ToS	Specifies the Type of Service level that the BIG-IP system assigns to TCP packets when sending them to clients.	0
Link QoS	Specifies the Quality of Service level that the BIG-IP system assigns to TCP packets when sending them to clients.	0
Selective ACKs	Specifies, when checked (enabled), that the system processes data using selective ACKs whenever possible, to improve system performance.	Enabled (Checked)
Extended Congestion Notification	Specifies, when checked (enabled), that the system uses the TCP flags CWR and ECE to notify its peer of congestion and congestion counter-measures.	Disabled (Cleared)
Extensions for High Performance (RFC 1323)	Specifies, when checked (enabled), that the system uses the timestamp and window scaling extensions for TCP (as specified in RFC 1323) to enhance high-speed network performance.	Enabled (Checked)
Limited Transmit Recovery	Specifies, when checked (enabled), that the system uses limited transmit recovery revisions for fast retransmits (as specified in RFC 3042) to reduce the recovery time for connections on a lossy network.	Enabled (Checked)
Slow Start	Specifies, when checked (enabled), that the system uses larger initial window sizes (as specified in RFC 3390) to help reduce round trip times.	Enabled (Checked)
Deferred Accept	Specifies, when checked (enabled), that the system defers allocation of the connection chain context until the system has received the payload from the client. Enabling this setting is useful in dealing with 3-way handshake denial-of-service attacks.	Disabled (Cleared)
Verified Accept	Specifies whether a SYN-ACK acknowledgement code is sent only if the server port is open. This option is not compatible with iRules.	Disabled (Cleared)
Nagle's Algorithm	Specifies, when checked (enabled), that the system applies Nagle's algorithm to reduce the number of short segments on the network. The default setting is disabled. Note that enabling this setting for interactive protocols such as telnet may cause degradation on high-latency networks.	Enabled (Checked)

Table 16.2 *Settings of a TCP profile*

Setting	Description	Default Value
Acknowledge on Push	Specifies, when enabled, significantly improved performance to Windows® and Mac OS® peers who are writing out on a very small send buffer.	Disabled (Cleared)
MD5 Signature	Specifies, when enabled, to use RFC2385 TCP-MD5 signatures to protect TCP traffic against intermediate tampering.	Disabled (Cleared)
MD5 Signature Passphrase	Specifies, when enabled, a plaintext passphrase which may be between 1 and 80 characters in length, and is used in a shared-secret scheme to implement the spoof-prevention parts of RFC2385.	No default value
Congestion Control	Specifies the algorithm to use to share network resources among competing users to reduce congestion.	High Speed
Congestion Metrics Cache	Specifies whether the system uses a cache for storing congestion metrics.	Enabled (Checked)
Appropriate Byte Counting (RFC 3465)	When Enabled , increases the congestion window by basing the increase amount on the number of previously unacknowledged bytes that each ACK covers. When this setting is disabled , in situations with lost ACK packets, the congestion window remains small for a longer period of time.	Enabled (Checked)
D-SACK (RFC 2883)	Specifies, when Enabled , the use of the selective ACK (SACK) option to acknowledge duplicate segments. If a peer does not send duplicate segments, the system disables SACK processing altogether. Note that when enabled, this setting requires more processing, to always populate the SACK with all duplicate segments.	Disabled (Cleared)
Packet Lost Ignore Rate	Specifies the threshold of packets lost per million at which the system performs congestion control. If you set the ignore rate to 10 and packet loss for a TCP connection is greater than 10 per million, congestion control occurs.	0
Packet Lost Ignore Burst	Specifies the probability of performing congestion control when multiple packets are lost, even if the Packet Lost Ignore Rate was not exceeded. The default means that the system performs congestion control if any packets are lost. Higher values decrease the chance of performing congestion control.	0
Initial Congestion Window Size	Specifies the initial congestion window size for connections to this destination. The actual window size is this value multiplied by the MSS (Maximal Segment Size) for the same connection. The default value means to use the values specified in RFC2414. The range is from 0 to 16 .	0

Table 16.2 Settings of a TCP profile

Setting	Description	Default Value
Initial Receive Window Size	Specifies the initial receive window size for connections to this destination. The actual window size is this value multiplied by the MSS (Maximal Segment Size) for the same connection. The default value means to use the Slow Start value. The range is from 0 to 16 .	0
Initial Retransmission Timeout Base Multiplier for SYN Retransmission	Specifies the initial RTO (Retransmission TimeOut) base multiplier for SYN retransmission, in milliseconds. This value is modified by the exponential backoff table to select the interval for subsequent retransmissions.	0
Delay Window Control	When Enabled , the system uses an estimate of queueing delay as a measure of congestion, in addition to the normal loss-based control, to control the amount of data sent.	Disabled (cleared)

Table 16.2 *Settings of a TCP profile*

For most of the TCP profile settings, the default values usually meet your needs. However, if the link that clients are using to access the virtual server is slow, or if server response time exceeds the request time of clients, you can increase the content spooling settings of the profile:

- **Proxy Buffer Low**
- **Proxy Buffer High**
- **Send Buffer**
- **Receive Window**

Increasing the byte values of these settings increases the amount of data that the BIG-IP system can buffer while waiting for a specific connection to accept that data.

◆ **Note**

*If you are using a TCP profile in a test environment, you can improve performance by disabling the **Slow Start**, **Bandwidth Delay**, and **Nagle's Algorithm** settings.*

Configuring a UDP profile

The UDP profile is a configuration tool for managing UDP network traffic. Table 16.3 lists and describes the settings of a UDP profile.

Setting	Description	Default Value
Name	This setting specifies a unique name for the profile.	No default value
Parent Profile	This setting specifies the profile that you want to use as the parent profile. Your new profile inherits all non-custom settings and values from the parent profile specified.	udp
Proxy Maximum Segment	This setting advertises the same maximum segment to the server as was negotiated with the client.	Disabled (Cleared)
Idle timeout	This setting specifies the number of seconds that a connection is idle before the connection flow is eligible for deletion.	60
IP ToS	This setting specifies the Type of Service level that the BIG-IP system assigns to UDP packets when sending them to clients.	0
Link QoS	This setting specifies the Quality of Service level that the BIG-IP system assigns to UDP packets when sending them to clients.	0
Datagram LB	This setting specifies, when checked (enabled), that the system load balances UDP traffic packet-by-packet.	Disabled (Cleared)
Allow No Payload	Specifies whether the system passes datagrams that contain header information, but no essential data.	Disabled (Cleared)

Table 16.3 *Settings of a UDP profile*



17

Using the Statistics Profile

- Introducing the Statistics profile
- Configuring a Statistics profile

Introducing the Statistics profile

In addition to the profiles described in previous chapters, you can configure the Statistics profile, which provides user-defined statistical counters.

For each profile type, the BIG-IP® system provides a pre-configured profile with default settings. In most cases, you can use these default profiles as is. If you want to change these settings, you can configure profile settings when you create a profile, or after profile creation by modifying the profile's settings.

For information on configuring other types of profiles, see the following:

- Chapter 14, *Managing HTTP and FTP Traffic*.
- Chapter 15, *Enabling Session Persistence*.
- Chapter 16, *Managing Protocol Profiles*.

Configuring a Statistics profile

The Statistics profile provides user-defined statistical counters. Each profile contains 32 settings (**Field1** through **Field32**), which define named counters. Using a Tcl-based iRule command, you can use the names to manipulate the counters while processing traffic.

For example, you can create a profile named **my_stats**, which assigns the counters **tot_users**, **cur_users**, and **max_users** to the profile settings **Field1**, **Field2**, and **Field3** respectively. You can then write an iRule named **track_users**, and then assign the **my_stats** profile and the **track_users** iRule to a virtual server named **stats-1**. Figure 17.1 shows this configuration.

```
profile stats my_stats {
    defaults from stats
    field1 tot_users
    field2 cur_users
    field3 max_users
}

rule track_users {
    when CLIENT_ACCEPTED {
        STATS::incr my_stats tot_users
        STATS::setmax my_stats max_users [STATS::incr my_stats
cur_users]
    }
}

virtual stats-1 {
    destination 10.10.55.66:http
    ip protocol tcp
    profile http my_stats tcp
    pool pool1
    rule track_users
}
```

Figure 17.1 Example of Statistics profile counters used in an iRule

In this example, the counter **tot_users** counts the total number of connections, the counter **cur_users** counts the current number of connections, and the counter **max_users** retains the largest value of the counter **cur_users**.

For information on iRules® **STATS** commands, see the F5 Networks DevCentral web site <http://devcentral.f5.com>, and Chapter 18, *Writing iRules*.



18

Writing iRules

- Introducing iRules
- Creating iRules
- Controlling iRule evaluation
- Using iRule commands
- Working with profiles
- Enabling session persistence with iRules
- Creating, managing, and using data groups

Introducing iRules

You can use **iRules**[®] to manage your network traffic within the BIG-IP[®] Link Controller[™] system. Using syntax based on the industry-standard Tools Command Language (Tcl), the iRules feature not only allows you to select pools based on header data, but also allows you to direct traffic by searching on any type of content data that you define. Thus, the iRules feature significantly enhances your ability to customize your content switching to suit your exact needs.

The remainder of this introduction presents an overview of iRules, lists the basic elements that make up an iRule, and shows some examples of how to use iRules to direct traffic to a specific destination such as a pool or a particular node.

Important

For complete and detailed information on iRules syntax, see F5 Networks DevCentral web site, <http://devcentral.f5.com>. For information on standard Tcl syntax, see <http://tmml.sourceforge.net/doc/tcl/index.html>.

What is an iRule?

An iRule is a script that you write if you want individual connections to target a pool other than the default pool defined for a virtual server. iRules allow you to more directly specify the destinations to which you want traffic to be directed. Using iRules, you can send traffic not only to pools, but also to individual pool members, ports, or URIs.

The iRules you create can be simple or sophisticated, depending on your content-switching needs. Figure 18.1 shows an example of a simple iRule.

```
when CLIENT_ACCEPTED {  
    if { [IP::addr [IP::client_addr] equals 10.10.10.10] } {  
        pool my_pool  
    }  
}
```

Figure 18.1 Example of an iRule

This iRule is triggered when a client-side connection has been accepted, causing the BIG-IP system to send the packet to the pool **my_pool**, if the client's address matches **10.10.10.10**.

The syntax that you use to write iRules is based on the Tool Command Language (Tcl) programming standard. Thus, you can use many of the standard Tcl commands, plus a robust set of extensions that the BIG-IP system provides to help you further increase load balancing efficiency.

Basic iRule elements

iRules are made up of these basic elements:

- Event declarations
- Operators
- iRule commands

Event declarations

iRules are event-driven, which means that the BIG-IP system triggers an iRule based on an event that you specify in the iRule. An *event declaration* is the specification of an event within an iRule that causes the BIG-IP system to trigger that iRule whenever that event occurs. An example of an event declaration that can trigger an iRule is **CLIENT_ACCEPTED**, which triggers an iRule when a client has established a connection as shown in Figure 18.2.

```
when CLIENT_ACCEPTED {  
    if { [IP::addr [IP::client_addr] equals 10.10.10.10 ] {  
        pool my_pool  
    }  
}
```

Figure 18.2 Example of an event declaration within an iRule

For more information on iRule events, see *Specifying events*, on page 18-6.

Operators

An iRule operator compares two operands in an expression. In addition to using the Tcl standard operators, you can use the operators listed in Table 18.1.

Operator	Syntax
Relational operators	contains matches equals starts_with ends_with matches_regex
Logical operators	not and or

Table 18.1 iRule operators

iRule commands

An *iRule command* within an iRule causes the BIG-IP system to take some action.

- ◆ **Statement commands** cause actions such as selecting a traffic destination or assigning a SNAT translation address. An example of a statement command is **pool <name>**, which directs traffic to the named load balancing pool. For more information, see *Using iRule commands*, on page 18-9.
- ◆ **Utility commands** are functions that are useful for parsing and manipulating content, such as the command in the following example.

```
when CLIENT_ACCEPTED {  
  if IP::client_addr matchclass $::good_addr {  
    pool current_data  
  } else {  
    pool old_data  
  }  
}
```

Figure 18.3 Example of the **pool** command within an iRule

Specifying traffic destinations and address translations

As described in the previous section, iRule commands instruct the BIG-IP system to take direct action in some way. The following sections show examples of iRule commands that either direct traffic to a specific destination or assign translation addresses for SNAT implementation.

Selecting a load balancing pool

Once you have specified a query within your iRule, you can use the **pool** command to select a load balancing pool to which you want the BIG-IP system to send a request. Figure 18.4 shows an example of this command.

```
when CLIENT_ACCEPTED {  
  if { [matchclass [IP::remote_addr] equals $::aol] }  
  {  
    pool aol_pool  
  } else {  
    pool all_pool  
  }  
}
```

Figure 18.4 Example of the **pool** command within an iRule

Selecting a specific server

As an alternative to the **pool** command, you can also write an iRule that directs traffic to a specific server. To do this, you use the **node** command. Figure 18.5 shows an example of this command.

```
when CLIENT_ACCEPTED {  
  if { TCP::port equals 80 } {  
    node 10.1.2.200 80  
  }  
}
```

Figure 18.5 Example of the **node** command within an iRule

Assigning translation addresses for SNAT connections

The iRules feature includes the two statement commands **snat** and **snatpool**. Using the **snat** command, you can assign a specified translation address to an original IP address from within the iRule, instead of using the SNAT screens within the Configuration utility, as shown in Figure 18.6.

Using the **snatpool** command also assigns a translation address to an original IP address, although unlike the **snat** command, the **snatpool** command causes the BIG-IP system to select the translation address from a specified SNAT pool that you previously created, as shown in Figure 18.7.

For more information on implementing SNATs, see Chapter 19, *Configuring SNATs and NATs*.

```
when CLIENT_ACCEPTED {  
  if { [IP::addr [IP::client_addr] equals 10.10.10.10] }  
  {  
    snat 192.168.1.100  
    pool my_pool  
  }  
}
```

Figure 18.6 Example of using the **snat** command to assign a SNAT translation IP address

```
when CLIENT_ACCEPTED {  
  if { [IP::addr [IP::client_addr] equals 10.10.10.10] }  
  {  
    snatpool my_snat_pool  
    pool my_pool  
  }  
}
```

Figure 18.7 Example of using the **snatpool** command to assign a SNAT translation address

Creating iRules

You create an iRule using the Configuration utility.

To create an iRule

1. On the Main tab, expand **Local Traffic**, and click **iRules**.
The iRules List screen opens.
2. In the upper right corner, click **Create**.
3. In the **Name** box, type a 1- to 31-character name.
4. In the **Definition** box, type the syntax for your iRule.
5. Click **Finished**.

For detailed syntax information on writing iRules, see the F5 Networks DevCentral web site <http://devcentral.f5.com>.

Important

*Once you have created an iRule, you need to configure a virtual server to reference the iRule. For information on configuring a virtual server to reference an iRule, see Chapter 6, **Configuring Virtual Servers**.*

Controlling iRule evaluation

In a basic system configuration where no iRule exists, the BIG-IP system directs incoming traffic to the default pool assigned to the virtual server that receives that traffic. However, you might want the BIG-IP system to direct certain kinds of connections to other destinations. The way to do this is to write an iRule that directs traffic to that other destination, contingent on a certain type of event occurring. Otherwise, traffic continues to go to the default pool assigned to the virtual server.

iRules are therefore evaluated whenever an event occurs that you have specified in the iRule. For example, if an iRule includes the event declaration **CLIENT_ACCEPTED**, then the iRule is triggered whenever the BIG-IP system accepts a client connection. The BIG-IP system then follows the directions in the remainder of the iRule to determine the destination of the packet.

Configuration prerequisites

Before the BIG-IP system can evaluate the iRule that you have written, you must do the following:

- ◆ **Assign the iRule to a virtual server.**
When an iRule is assigned to virtual server, this means that the virtual server *references* the iRule, similar to the way that a virtual server references a pool or a profile.
- ◆ **Ensure that the virtual server references a TCP or UDP profile.**

For information on assigning iRules and profiles to virtual servers, see Chapter 6, *Configuring Virtual Servers*.

Specifying events

The iRules feature includes several types of event declarations that you can make in an iRule. Specifying an event declaration determines when the BIG-IP system evaluates the iRule. The following sections list and describe these event types. Also described is the concept of iRule context and the use of the **when** keyword.

Event types

The iRule command syntax includes several types of event declarations that you can specify within an iRule. For example, global events, such as **CLIENT_ACCEPTED**.

For a complete list of iRule events and their descriptions, see F5 Networks DevCentral web site, <http://devcentral.f5.com>.

iRule context

For every event that you specify within an iRule, you can also specify a context, denoted by the keywords **clientside** or **serverside**. Because each event has a default context associated with it, you need only declare a context if you want to change the context from the default.

For example, Figure 18.8 shows **my_iRule1**, which includes the event declaration **CLIENT_ACCEPTED**, as well as the iRule command **IP::remote_addr**. In this case, the IP address that the iRule command returns is that of the client, because the default context of the event declaration **CLIENT_ACCEPTED** is **clientside**.

```
when CLIENT_ACCEPTED {
  if { [IP::addr [IP::remote_addr] equals 10.1.1.80] } {
    pool my_pool1
  }
}
```

Figure 18.8 An iRule that uses default clientside context

Similarly, if you include the event declaration **SERVER_CONNECTED** in an iRule as well as the iRule command **IP::remote_addr**, the IP address that the iRule command returns is that of the server, because the default context of the event declaration **SERVER_CONNECTED** is **serverside**.

Figure 18.8 shows what happens when you write an iRule that uses the default context when processing iRule commands. You can, however, explicitly specify the **clientside** and **serverside** keywords to alter the behavior of iRule commands.

Continuing with the previous example, Figure 18.9 shows the event declaration **SERVER_CONNECTED** and explicitly specifies the **clientside** keyword for the iRule command **IP::remote_addr**. In this case, the IP address that the iRule command returns is that of the client, despite the serverside default context of the event declaration..

```
when SERVER_CONNECTED {
  if {[IP::addr [IP::addr [clientside {IP::remote_addr}]]
    equals 10.1.1.80] } {
    discard
  }
}
```

Figure 18.9 An iRule that explicitly specifies context

Using the when keyword

You make an event declaration in an iRule by using the **when** keyword, followed by the event name. The previous figure shows an example of an event declaration in an iRule.

Listing iRules on a virtual server

When you assign multiple iRules as resources for a virtual server, it is important to consider the order in which you list them on the virtual server. This is because the BIG-IP system processes duplicate iRule events in the order that the applicable iRules are listed. An iRule event can therefore terminate the triggering of events, thus preventing the BIG-IP system from triggering subsequent events.

◆ Note

If an iRule references a profile, the BIG-IP system processes this type of iRule last, regardless of its order in the list of iRules assigned to a virtual server.

Using iRule commands

There are three kinds of iRule commands:

- Statement commands
- Query and manipulation commands
- Utility commands (also known as functions)

Statement commands

Some of the commands available for use within iRules are known as statement commands. ***Statement commands*** enable the BIG-IP system to perform a variety of different actions. For example, some of these commands specify the pools or servers to which you want the BIG-IP system to direct traffic. Other commands specify translation addresses for implementing SNAT connections. Still others specify objects such as data groups or a persistence profiles.

For a complete list of statement commands, see F5 Networks DevCentral web site, <http://devcentral.f5.com>.

Query and manipulation commands

Using iRules commands, you can query for specific data contained in the header or content of a request or response, or you can manipulate that data. Data manipulation refers to inserting, replacing, and removing data, as well as setting certain values found in headers and cookies.

For example, using the **IP::idle_timeout** command within in iRule, you can query for the current idle timeout value that is set in a packet header and then load balance the packet accordingly. You can also use the **IP::idle_timeout** command to set the idle timeout to a specific value of your choice.

iRule query and manipulation commands are grouped into categories called ***namespaces***. Except for commands in the global namespace, each iRule query or manipulation command includes the namespace in its command name. For example, one of the commands in the **IP** namespace is **IP::idle_timeout**.

For a complete list of namespaces for iRules commands, see F5 Networks DevCentral web site, <http://devcentral.f5.com>.

Utility commands

The BIG-IP system includes a number of utility commands that you can use within iRules. You can use these commands to parse and retrieve content, encode data into ASCII format, verify data integrity, and retrieve information about active pools and pool members.

Working with profiles

When you are writing an iRule, you might want that iRule to know the value of a particular profile setting so that it can make a more-informed traffic management decision. Fortunately, the iRules feature includes a command that is specifically designed to read the value of profile settings that you specify within the iRule.

Not only can iRules read the values of profile settings, but they can also override values for certain settings. This means that you can apply configuration values to individual connections that differ from the values the BIG-IP system applies to most connections passing through a virtual server.

Reading profile settings

The iRules feature includes a command called **PROFILE**. When you specify the **PROFILE** command in an iRule and name a profile type and setting, the iRule reads the value of that particular profile setting. To do this, the iRule finds the named profile type that is assigned to the virtual server and reads the value of the setting that you specified in the **PROFILE** command sequence. The iRule can then use this information to manage traffic.

For example, you can specify the command **PROFILE::tcp idle_timeout** within your iRule. The BIG-IP system then finds the TCP profile that is assigned to the virtual server (for example, **my_tcp**) and queries for the value that you assigned to the **Idle Timeout** setting.

Overriding profile settings

Some of the iRule commands for querying and manipulating header and content data have equivalent settings within various profiles. When you use those commands in an iRule, as shown in Figure 18.10, and an event triggers that iRule, the BIG-IP system overrides the values of those profile settings, using the value specified within the iRule instead.

```
when CLIENT_ACCEPTED {
  if { [IP::tos]==16} {
    pool telnet pool
  }
  else {
    pool slow pool
  }
}
```

Figure 18.10 Example of overriding the pool setting in a profile

Enabling session persistence with iRules

Chapter 15, *Enabling Session Persistence*, describes how to enable session persistence by configuring a persistence profile and assigning it to a virtual server. As described in that chapter, the BIG-IP system applies those persistence profile settings to every applicable session that passes through the virtual server.

The BIG-IP system includes a special iRule command, **persist**, for implementing the types of session persistence described in Chapter 15, *Enabling Session Persistence*. You simply type the **persist** command in your iRule, specifying a persistence type.

You can use the **persist none**, **srcaddr**, and **destaddr** commands in any circumstance, even if a corresponding persistence profile is not configured and assigned to the virtual server. For information on assigning a persistence profile to a virtual server, see Chapter 6, *Configuring Virtual Servers*.

Creating, managing, and using data groups

Data groups are useful when writing iRules. A **data group** is simply a group of related elements, such as a set of IP addresses for AOL clients. When you specify a data group along with the **matchclass** command or the **contains** operator, you eliminate the need to list multiple values as arguments in an iRule expression

To understand the usefulness of data groups, it is helpful to first understand the **matchclass** command and the **contains** operator.

Using the matchclass command

The BIG-IP system includes an iRule command called **matchclass**, which you can use to select a pool based on whether the command being used in the iRule represents a member of a specific data group. When you use the **matchclass** command, the BIG-IP system knows that the string following the identifier is the name of a data group.

For example, using the **matchclass** command, you can cause the BIG-IP system to load balance all incoming AOL connections to the pool **aol_pool**, if the value of the **IP::remote_addr** command is a member of the data group AOL. Figure 18.11, on page 18-12 shows this type of iRule. In this case, the **matchclass** command simply indicates that the object named **aol** is a collection of values (that is, a data group).

```
when CLIENT_ACCEPTED {  
    if { [matchclass [IP::remote_addr] equals $::aol] } {  
        pool aol_pool  
    } else {  
        pool all_pool  
    }  
}
```

Figure 18.11 An iRule based on the `matchclass` command

Note that an expression such as `[IP::remote__addr] equals matchclass $::aol` is true if the expression is true with at least one specific value in the data group.

Creating data groups

When using the **matchclass** command within an iRule, you can specify any of three types of data groups:

- Addresses data group - A collection of IP addresses
- String data group - A collection of strings, such as `*.jpg`
- Integer data group - A collection of numeric values.

The following sections describe these data group types.

◆ **Note**

The size of a data group is limited by system resources only.

Address data groups

There are two types of IP address data groups, network IP address and host IP address.

The following procedure creates a network or host address data group:

To create an address data group

1. On the Main tab, expand **Local Traffic**, and click **iRules**.
The iRules List screen opens.
2. On the menu bar, click **Data Group List**.
3. In the upper right corner of the screen, click **Create**.
4. In the **Name** box, type a unique name for the data group, such as **my_address_group**.
5. In the **Type** box, select **Address**.
The screen expands to show more settings.
6. In the Records section, select **Host** or **Network**.

7. In the **Address** box, type the first IP address for the data group. If you are creating a network data group, also enter a network mask in the **Mask** box.
8. Click **Add**.
The entry displays in the **Address Records** box.
9. Repeat steps 7 and 8 until you have entered all IP addresses.
10. Click **Finished**.

String data groups

A string data group contains a list of strings, such as ***.jpg** or ***.gif**. The following procedure creates a string data group.

Note that this example shows the use of escape characters for the quotation marks.

To create a string data group

1. On the Main tab, expand **Local Traffic**, and click **iRules**.
The iRules List screen opens.
2. On the menu bar, click **Data Group List**.
3. In the upper right corner of the screen, click **Create**.
4. In the **Name** box, type a unique name for the data group, such as **my__images**.
5. In the **Type** box, select **String**.
The screen expands to show the string-specific settings.
6. In the **String** box, type the first string for the data group.
7. Click **Add**.
The entry displays in the **String Records** box.
8. Repeat steps 6 and 7 until you have entered all strings.
9. Click **Finished**.

Integer data groups

An integer data group contains a list of integers. The following procedure describes how to create an integer data group.

To create an integer data group

1. On the Main tab, expand **Local Traffic**, click **iRules**.
The iRules List screen opens.
2. On the menu bar, click **Data Group List**.
3. In the upper right corner of the screen, click **Create**.

4. In the **Name** box, type a unique name for the data group, such as **my__integer_group**.
5. In the **Type** box, select **Integer**.
The screen expands to display the Records section.
6. In the **Integer** box, type the first integer for the data group.
7. Click **Add**.
The entry displays in the **Integer Records** box.
8. Repeat steps 6 and 7 until you have added all integers.
9. Click **Finished**.

Storage options

The BIG-IP system allows you to store data groups in two ways, either in-line or externally.

In-line storage

When you create data groups, the BIG-IP system automatically saves them in their entirety in the **bigip.conf** file. This type of storage is known as *in-line storage*.

When any data in the data group needs to be updated, the entire data group must be reloaded. In general, in-line storage uses additional system resources due to extensive searching requirements on large data groups. Also, in-line storage requires you to reload entire data groups when incrementally updating data. For these reasons, the BIG-IP system offers you the ability to store your data groups externally, that is, outside of the **bigip.conf** file.

External storage

You have the option to store data groups in another location on the BIG-IP system, that is, outside of the **bigip.conf** file. Such data groups are called *external data groups*. The default location for storing external data groups is the **/config** directory. Because the data group is stored externally in another location, the **bigip.conf** file itself contains only meta-data for the data group. The data in an externally-stored data group file is stored as a comma-separated list of values (CSV format).

Creating external data groups is useful because data does not need to be sorted when being loaded. Instead, data is stored in a hash-table in the kernel. This storage method translates to improvements in performance when an iRule uses a large data group to direct traffic to a pool.

To store data groups externally

1. On the Main tab, expand **Local Traffic**, and click **iRules**.
The iRules List screen opens.
2. On the menu bar, click **Data Group List**.
3. In the upper right corner of the screen, click **Create**.
4. In the **Name** box, type the name of the existing data group that you want to store in an external location.
5. In the **Type** box, select **(External File)**.
The screen expands to display the Records section.
6. Specify a storage location:
 - If you do not want to store your data group in the default external location (**/config**), use the **Path / Filename** box to specify a path name and file name for the external location, for example, **/home/my_address_group**.
This file name should match the name that you assigned to the data group itself.
 - If you want to store your data group in the default external location (**/config**), leave the **Path / Filename** box empty.
7. In the **File Contents** box, select the file type that pertains to the data group (**Address**, **String**, or **Integer**).
8. Click **Finished**.

The BIG-IP system stores the data in an external data group file in comma-separated lists, and the formats of any data values, such as IP addresses, match the formats used in the **bigip.conf** file. Figure 18.12 shows the contents of the data group file **/home/ip2.data group**.

```
network 195.93.32.0 mask 255.255.255.0,
network 195.93.33.0 mask 255.255.255.0,
network 195.93.34.0 mask 255.255.255.0,
network 195.93.48.0 mask 255.255.255.0,
network 195.93.49.0 mask 255.255.255.0,
network 195.93.50.0 mask 255.255.255.0
```

Figure 18.12 An example of an external data group file

Displaying data group properties

Using the Configuration utility, you can display the properties of an existing data group.

To display the properties of an data group

1. On the Main tab, expand **Local Traffic**, and click **iRules**.
The iRules List screen opens.
2. On the menu bar, click **Data Group List**.
3. Click the name of a data group.
The properties of the data group display.

Managing data group members

Using the Configuration utility, you can add members to or delete members from an existing data group.

To add members to a data group

1. On the Main tab, expand **Local Traffic**, and click **iRules**.
The iRules List screen opens.
2. On the menu bar, click **Data Group List**.
3. Click the name of a data group.
The properties of the data group display.
4. In the records area, type an address, string, or integer in the appropriate box.
5. Click **Add**.
6. Click **Update**.



19

Configuring SNATs and NATs

- Introducing secure network address translation
- Creating a SNAT pool
- Implementing a SNAT
- Implementing a NAT
- Managing SNATs and NATs
- SNAT examples

Introducing secure network address translation

A virtual server configured on a BIG-IP® Link Controller™ system translates the destination IP address of an incoming packet to another destination IP address, for the purpose of load balancing that packet. Normally, the source IP address remains unchanged.

As an option, you can also create a secure network address translation (SNAT). A **SNAT** is an object that maps an original client IP address (that is, a source IP address) to a translation address that you choose. Thus, a SNAT causes the BIG-IP system to translate the source IP address of an incoming packet to an address that you specify. The purpose of a SNAT is simple: to ensure that the target server sends its response back through the BIG-IP system rather than to the original client IP address directly.

To create a SNAT, you either use the Configuration utility or write an iRule, depending on the type of SNAT you are creating. For information on iRules®, see the F5 Networks DevCentral web site <http://devcentral.f5.com>, or Chapter 18, *Writing iRules*.

◆ Note

This type of translation has no effect on the destination address translation that a virtual server performs.

Examples of scenarios where a SNAT is useful are:

- To connect to an external device that requires a routable return IP address
- To connect to a virtual server with a node that is on the same IP subnet as the client

◆ Tip

*Because the purpose of a SNAT is simply to change the source IP address of incoming packets, the term **secure network address translation** is a slight misnomer. A better way to define the SNAT acronym would be **source network address translation**, or **source NAT**.*

How does a SNAT work?

A SNAT works in the following way:

1. The BIG-IP system receives a packet from an original client IP address and checks to see if that source address is defined in a SNAT.
2. If the client's IP address is defined in a SNAT, the BIG-IP system changes that source IP address to the translation address defined in the SNAT.
3. The BIG-IP system then sends the client request, with the SNAT translation address as the source address, to the target server.

The end result of this process is that the target server has a routable IP address for the client that the server can specify as the destination IP address in its response.

Mapping original IP addresses to translation addresses

When you create a SNAT, you map an original IP address to a translation address in one of several ways, depending on your needs. For example, you can explicitly map an original IP address to a single translation address, or you can create a *pool* of translation addresses and map the original IP address to that pool of addresses.

Mapping a specific original IP address to a specific translation address

One way to create a SNAT is to directly map one or more original IP addresses to a specific translation address that you choose. A SNAT that you create in this way is a type of standard SNAT. A ***standard SNAT*** is a SNAT object that you create using the New SNAT screen of the Configuration utility. For more information on standard SNATs, see *Implementing a SNAT*, on page 19-5.

Using the SNAT automap feature

Another way to create a SNAT is to use a feature of the BIG-IP system called SNAT automap. The ***SNAT automap*** feature automatically maps one of the system's self IP addresses to the original IP address you specify during SNAT creation. When you use this feature, you do not need to explicitly specify a translation address.

A SNAT that you create in this way is a type of standard SNAT. For more information on standard SNATs, see *Implementing a SNAT*, on page 19-5.

Mapping a specific original IP address to a pool of translation addresses

You can also create a SNAT by creating a pool of translation addresses and then mapping an original IP address to the entire translation pool. This pool of translation addresses is known as a **SNAT pool**. You create a SNAT pool using the New SNAT Pool screen of the Configuration utility. For information on creating a SNAT pool, see *Implementing a SNAT*, on page 19-5.

Once you have created a SNAT pool and mapped it to an original IP address, and the virtual server then receives a packet from the original IP address, the BIG-IP system chooses a translation address from that SNAT pool. The system then translates the original IP address to the chosen address.

You can map an original IP address to the SNAT pool in one of two ways:

- ◆ **By creating a SNAT object.**

A SNAT that you create this way, using the New SNAT screen in the Configuration utility, is a type of standard SNAT. For more information on standard SNATs, see *Creating a standard SNAT*, on page 19-6.

- ◆ **By writing an iRule.**

In this case, you do not create a SNAT object. Instead, you write an iRule that includes a **snat** or **snatpool** command. The type of SNAT that you create by writing an iRule is called an intelligent SNAT. An **intelligent SNAT** is the mapping of one or more original client IP addresses to a translation address through the use of an iRule. For more information on intelligent SNATs, see *Creating an intelligent SNAT*, on page 19-9.

Mapping all original IP addresses to a pool of translation addresses

Yet another way to create a SNAT is to create a SNAT pool (using the New SNAT Pool screen of the Configuration utility) and directly assign it to a virtual server as a resource of that virtual server. Once you have assigned a SNAT pool to a virtual server, the BIG-IP system automatically maps all original IP addresses coming through the virtual server to that SNAT pool. As with intelligent SNATs, you do not create a SNAT object, with the New SNAT screen, in the Configuration utility. For more information on this type of SNAT, see *Assigning a SNAT pool directly to a virtual server*, on page 19-9.

Creating a SNAT pool

If you decide to use a SNAT pool as the way to specify translation addresses in your SNAT, you must first create the SNAT pool, specifying one or more translation addresses that you want to include in the SNAT pool. You create a SNAT pool using the Configuration utility. For background information on SNAT pools, see *Mapping a specific original IP address to a pool of translation addresses*, on page 19-3.

After creating the SNAT pool, you then create the type of SNAT that best suits your needs (a standard SNAT, an intelligent SNAT, or a SNAT pool that you assign directly to a virtual server). To understand the different types of SNATs that you can create, see *Implementing a SNAT*, on page 19-5.

A SNAT pool has two settings that you must configure when you create it. Table 19.1 lists and describes these settings.

Property	Description	Default Value
Name	The unique name of the SNAT pool.	No default value
Member List	The list of IP addresses that you want to include in SNAT pool. If the IP addresses that you add are not already designated as translation addresses, the BIG-IP system automatically designates them as such and assigns them the appropriate properties with their default values. This setting is required.	No default value

Table 19.1 *Properties of a SNAT pool*

Each translation address that you add to the SNAT pool has settings that you can configure after you add the address to the SNAT pool. For information on these settings, see *Specifying a translation address*, on page 19-7.

Once you create a SNAT pool, you must do one of the following:

- Reference the SNAT pool from within a SNAT object that you create. You do this when you create a standard SNAT. For more information, see *Creating a standard SNAT*, on page 19-6.
- Reference the SNAT pool from within an iRule and then assign the iRule to a virtual server as a resource. You do this when you create an intelligent SNAT. For more information, see *Creating an intelligent SNAT*, on page 19-9.
- Assign the SNAT pool directly to a virtual server as a resource. For more information, see *Assigning a SNAT pool directly to a virtual server*, on page 19-9.

To create a SNAT pool

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **SNAT Pool List**.
The list of existing SNAT pools displays.
3. In the upper-right corner of the screen, click **Create**.
4. For the **Name** setting, type a unique name for the SNAT pool.
5. For the **Member List** setting, type an IP address.
6. Click **Add**.
7. Repeat steps 5 and 6 for each translation address that you want to add.
8. Click **Finished**.

Implementing a SNAT

Before implementing secure network address translation, you should decide which type of SNAT you want to create. The types of SNATs you can create are:

◆ **Standard SNAT**

A *standard SNAT* is an object you create, using the Configuration utility, that specifies the mapping of one or more original client IP addresses to a translation address. For this type of SNAT, the criteria that the BIG-IP system uses to decide when to apply the translation address is based strictly on the original IP address. That is, if a packet arrives from the original IP address that you specified in the SNAT, then the BIG-IP system translates that address to the specified translation address.

There are three types of standard SNATs that you can create:

- A SNAT in which you specify a specific translation address
- A SNAT that uses the automap feature
- A SNAT in which you specify a SNAT pool as your translation address

◆ **Intelligent SNAT**

Like a standard SNAT, an *intelligent SNAT* is the mapping of one or more original client IP addresses to a translation address. However, you implement this type of SNAT mapping within an iRule instead of by creating a SNAT object. For this type of SNAT, the criteria that the BIG-IP system uses to decide when to apply a translation address is based on any piece of data you specify within the iRule, such as an HTTP cookie or a server port.

◆ **SNAT pool assigned as a virtual server resource**

This type of SNAT consists of just a SNAT pool that you directly assign as a resource to a virtual server. When you implement this type of SNAT, you create a SNAT pool only; you do not need to create a SNAT object or an iRule.

For more information on mapping original IP addresses to translation addresses, see *Mapping original IP addresses to translation addresses*, on page 19-2.

Creating a standard SNAT

You create a standard SNAT using the Configuration utility. The translation address or addresses that you map to an original IP address can be either a specific IP address, an existing SNAT pool, or a self IP address (using the automap feature).

When you create a standard SNAT, the BIG-IP system automatically assigns a set of properties to the SNAT. While you must configure the **Name** and **Translation** settings at the time that you create the SNAT, you can use the default values for the other settings, or modify those values later.

To create a standard SNAT

1. On the Main tab, expand **Local Traffic**, and click **SNATs**. The SNAT List screen opens.
2. In the upper-right corner of the screen, click **Create**.
3. For the **Name** setting, type a unique name for the SNAT.
4. For the **Translation** setting, select **IP Address**, **SNAT Pool**, or **Automap**.
5. If you selected **IP Address** or **SNAT Pool**, type an IP address or select a SNAT pool name.
6. Change or retain all other values.
7. Click **Finished**.

Table 19.2 shows the settings that you can configure for a SNAT. Following the table are detailed descriptions of each setting.

Property	Description	Default Value
Name	Specifies the unique name of the standard SNAT. Setting this property is required.	No default value
Translation	Depending on the value selected, specifies an individual IP address, a SNAT pool name, or the Automap option. Possible values are: IP Address , SNAT Pool , or Automap .	Automap
Origin	Specifies the original client IP addresses to which you want to map a translation address or pool of translation or self IP addresses. Possible values are All Addresses or Address List .	All Addresses
VLAN Traffic	The VLAN to which you want the SNAT to apply. Possible values are: ALL VLANS , Enabled On , and Disabled On .	ALL VLANS
Auto Last Hop	Specifies whether the system automatically maps the last hop for pools.	Enabled (checked)

Table 19.2 *Properties of a standard SNAT*

Specifying a SNAT name

The most basic setting you can configure for a standard SNAT is the SNAT name. SNAT names are case-sensitive and may contain letters, numbers, and underscores (_) only. Reserved keywords are not allowed.

Each SNAT that you define must have a unique name.

Specifying a translation address

The **Translation** setting specifies the translation addresses that you want to map to your original client IP addresses. For background information on translation addresses, see *Mapping original IP addresses to translation addresses*, on page 19-2.

There are three possible values for the **Translation** setting:

◆ IP Address

When creating a SNAT, you can specify a particular IP address that you want the SNAT to use as a translation address. For the procedure on specifying a particular translation address, see *To explicitly define a translation address*, on page 19-13.

◆ SNAT pool

Specifying this value allows you to specify an existing SNAT pool to which you want to map your original client IP address. For information on SNAT pools and how to create them, see *Creating a SNAT pool*, on page 19-4. For an example of a standard SNAT that uses a SNAT pool, see *Example 1 - Establishing a standard SNAT that uses a SNAT pool*, on page 19-15.

◆ **Automap**

Similar to a SNAT pool, the SNAT automap feature allows you to map one or more original client IP addresses to a pool of translation addresses. However, with the SNAT automap feature, you do not need to create the pool. Instead, the BIG-IP system effectively creates a pool for you, using all of the BIG-IP system's self IP addresses as the translation addresses for the pool.

When you specify a translation address or a SNAT pool, the BIG-IP system automatically assigns a set of properties to that translation address. You can use the default values for these properties, or you can change them to suit your needs. Table 19.3 lists and describes the properties of a translation address.

Property	Description	Default Value
IP address	The IP address that you want to designate as a translation address. This is a required setting.	No default value
State	The state of the translation address, that is, enabled or disabled. If set to disabled , the translation address is not used to initiate a connection.	Enabled
ARP	A setting that determines whether or not the BIG-IP system responds to ARP requests or sends gratuitous ARPs.	Enabled
Connection Limit	A limit on the number of connections a translation address must reach before it no longer initiates a connection. The default value of 0 indicates that the setting is disabled.	0
TCP Idle Timeout	A timer that defines the number of seconds that TCP connections initiated using a SNAT address are allowed to remain idle before being automatically disconnected. Possible values are Indefinite or Specify .	Indefinite
UDP Idle Timeout	A timer that defines the number of seconds that UDP connections initiated using a SNAT address are allowed to remain idle before being automatically disconnected. Possible values are Indefinite or Specify .	Indefinite
IP Idle Timeout	A timer that defines the number of seconds that IP connections initiated using a SNAT address are allowed to remain idle before being automatically disconnected. Possible values are Indefinite or Specify .	Indefinite

Table 19.3 *Properties of a SNAT translation address*

Specifying original IP addresses

The **Origin** setting specifies the original client IP addresses that you want to map to translation addresses. You can add one IP address or multiple IP addresses as values for this setting.

Specifying VLAN traffic

The **VLAN Traffic** setting specifies the VLANs to which you want the SNAT to apply. Possible values are: **ALL VLANS**, **Enabled On**, and **Disabled On**.

Creating an intelligent SNAT

One way to perform secure address translation is to create an intelligent SNAT. As described previously, an *intelligent SNAT* is not a SNAT object, but instead an iRule that maps of one or more original client IP addresses to a translation address. To create an intelligent SNAT, you must complete these tasks:

- If you are mapping an original IP address to a SNAT pool (as opposed to an individual translation address), use the New SNAT Pools screen to create one or more SNAT pools that include those translation addresses as members. For more information, see *To create a SNAT pool*, on page 19-5.
- Use the New iRule screen to create an iRule that includes the **snat** or **snatpool** command. These iRule commands specify the translation address or the pool of translation addresses that the BIG-IP system should use to select a translation address. For more information on iRules, see the F5 Networks DevCentral web site <http://devcentral.f5.com>, and Chapter 18, *Writing iRules*.
- From the Resources screen for the appropriate virtual server, assign the iRule as a resource to the virtual server. For more information on virtual servers, see Chapter 6, *Configuring Virtual Servers*.

◆ Note

*For an example of an intelligent SNAT, see **Example 2 - Establishing an intelligent SNAT**, on page 19-16.*

Assigning a SNAT pool directly to a virtual server

Rather than creating a SNAT object, or an intelligent SNAT using an iRule, you have the option of simply creating a SNAT pool and then assigning it as a resource directly to a virtual server. This eliminates the need for you to explicitly define original IP addresses to which to map translation addresses.

Implementing a NAT

A **network translation address** (NAT) provides an alias IP address that a node can use as its source IP address when making or receiving connections to clients on the external network. (This distinguishes it from a SNAT, which can initiate but not receive a connection.)

The IP addresses that identify nodes on the internal network need not be routable on the external network. This protects nodes from illegal connection attempts, but it also prevents nodes (and other hosts on the internal network) from receiving direct administrative connections, or from initiating connections to external servers, such as mail servers or databases.

Using NATs solves this problem. NATs assign to a particular node a routable IP address that the node can use as its source IP address when connecting to external servers. You can use the NAT IP address to connect directly to the node through the BIG-IP system, rather than having the BIG-IP system send the traffic to a random node according to the specified load balancing method.

◆ Note

Note that NATs do not support port translation, and are not appropriate for protocols that embed IP addresses in the packet, such as FTP, NT Domain or CORBA IIOP.

You must create a separate NAT for each node, using the Configuration utility. When you create a NAT, you configure a set of properties. While you must configure the **NAT Address** and **Origin Address** settings at the time that you create the NAT, you can use the default values for the other settings, or modify those values later.

To create a NAT

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **NAT List**.
3. In the upper right corner, click **Create**.
The New NAT screen opens.
4. In the **NAT Address** box, type the IP address that you want to use as a translation address.
5. In the **Origin Address** box, type the original client IP address to be translated.
6. Retain or modify all other values as necessary.
7. Click **Finished**.

Table 19.4 shows the settings that you can configure for a NAT, with a description of each.

NAT Attribute	Description	Default Value
NAT Address	An IP address that is routable on the external network of the BIG-IP system.	No default value
Origin Address	The original address is the node IP address of a host that you want to be able to connect to through the NAT.	No default value
State	The state of the NAT, that is, whether the NAT is enabled or disabled.	Enabled
ARP	A setting that instructs the BIG-IP system to respond to ARP requests from the specified NAT address, and send gratuitous ARP requests for router table updates.	Enabled
VLAN Traffic	VLANs to which the NAT is not to be mapped can be explicitly disabled, as when there is more than one internal VLAN.	All VLANs
Auto Last Hop	Specifies whether the system automatically maps the last hop for pools.	Default

Table 19.4 NAT configuration settings

In addition to these options, you can set up forwarding virtual servers that allow you to selectively forward traffic to specific addresses.

Additional restrictions

When using a NAT, you should be aware of the following restrictions:

- The IP address defined in the **Origin Address** box must be routable to a specific server behind the system.
- You must delete a NAT before you can redefine it.

Managing SNATs and NATs

Using the Configuration utility, you can manage existing SNATs in many ways. For example, you might want to view a list of existing SNAT pools before creating a new one. Or you might want to modify the way that a standard SNAT maps an original IP address to a translation address.

That tasks that you can perform when managing SNATs are:

- Viewing or modify a SNAT or NAT, or a SNAT pool
- Defining or viewing translation addresses
- Deleting SNATs or NATs, SNAT pools, and translation addresses
- Enabling or disabling SNATs or NATs for a load balancing pool
- Enabling or disabling SNAT or NAT translation addresses

Viewing or modifying SNATs, NATs, and SNAT pools

You can view or modify any SNATs, NATs, or SNAT pools that you created previously.

To view or modify a SNAT or NAT

1. On the Main tab, expand **Local Traffic**, and click **SNATs**. The SNAT List screen opens.
2. Select the type of item you want to view or modify:
 - If you want to view or modify a SNAT, click a SNAT name.
 - If you want to view or modify a NAT, click **NAT List** on the menu bar, and then click a NAT address.
3. View or modify the displayed settings.
4. If you modified any settings, click **Update**.

To view or modify a SNAT pool

1. On the Main tab, expand **Local Traffic**, and click **SNATs**. The SNAT List screen opens.
2. On the menu bar, click **SNAT Pool List**. The list of existing SNAT pools displays.
3. Click a SNAT pool name.
4. View or modify the displayed settings.
5. If you modified any settings, click **Update**.

Defining and viewing translation addresses

You can define a translation address or view any existing translation addresses the you defined previously.

To explicitly define a translation address

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **SNAT Translation List**.
3. In the upper-right corner of the screen, click **Create**.
4. Retain or change all property settings.
5. Click **Finished**.

To view translation addresses

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **SNAT Translation List**.
A list of existing translation addresses displays.
3. Click a translation address.
4. View or modify the displayed settings.
5. If you modified any settings, click **Update**.

Deleting SNATs, NATs, SNAT pools, and translation addresses

You can delete any existing SNAT, NAT, SNAT pool, or translation address that you created previously.

To delete a SNAT or a NAT

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. Select the type of item you want to delete:
 - If you want to delete a SNAT, locate the SNAT you want to delete, and check the Select box on the left.
 - If you want to delete a NAT, click **NAT List** on the menu bar, locate the NAT you want to delete, and check the Select box to the left.
3. At the bottom of the screen, click **Delete**.

To delete a SNAT pool

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **SNAT Pool List**.
The list of existing SNAT pools displays.
3. Locate the SNAT pool you want to delete, and check the Select box to the left.
4. At the bottom of the screen, click **Delete**.

To delete a translation address

1. On the Main tab, expand **Local Traffic**, and click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **SNAT Translation List**.
The list of existing translation addresses displays.
3. Locate the translation address you want to delete, and check the Select box to the left.
4. At the bottom of the screen, click **Delete**.

Enabling or disabling SNATs or NATs for a load balancing pool

When configuring a load balancing pool, you can specifically disable SNAT or NAT translations on any connections that use that pool. By default, this setting is enabled. For more information, see Chapter 5, *Configuring Load Balancing Pools*.

Enabling or disabling SNAT translation addresses

Using the Configuration utility, you can enable or disable an individual SNAT translation address.

To enable or disable a SNAT translation address

1. On the Main tab, expand **Local Traffic**, click **SNATs**.
The SNAT List screen opens.
2. On the menu bar, click **SNAT Translation List**.
3. Click the name of the address you want to enable or disable.
The properties screen for the SNAT opens.
4. From the **State** setting, select either **Enabled** or **Disabled**.
5. Click the **Update** button.

SNAT examples

The following examples demonstrate ways to implement SNATs that make use of SNAT pools. The examples illustrate how you can:

- Establish a standard SNAT that uses a SNAT pool
- Establish an intelligent SNAT

◆ Note

*To best illustrate SNATs that use SNAT pools, the following examples show sample entries from the BIG-IP system's **bigip.conf** file. Entries in the **bigip.conf** file represent the result of using the Configuration utility to configure the BIG-IP system.*

Example 1 - Establishing a standard SNAT that uses a SNAT pool

In some cases, you might need to create a SNAT that maps an original IP address to a SNAT pool instead of to an individual translation address. To illustrate this type of SNAT, suppose an ISP wants to provide two customers with two routable IP addresses each, for links to the Internet. The customers need to use these routable IP addresses as virtual IP addresses for inbound traffic to their own servers, and as translation addresses for outbound traffic from their servers.

In this case, the SNAT provides the solution. To implement the SNAT, the ISP takes the following three steps.

First, the ISP creates the load balancing pool **isp_pool**, shown in Figure 19.1.

```
pool isp_pool {  
  lb_method rr  
  member 199.5.6.254:0  
  member 207.8.9.254:0  
}
```

Figure 19.1 *bigip.conf* entries for a basic load balancing pool

Next, the ISP creates three SNAT pools: **customer1_snatpool**, **customer2_snatpool**, and **other_snatpool**. This is shown in Figure 19.2. Note that the BIG-IP system automatically designates the SNAT pool members as translation addresses.

```
snatpool customer1_snatpool {  
  member 199.5.6.10  
  member 207.8.9.10  
}  
snatpool customer2_snatpool {  
  member 199.5.6.20  
  member 207.8.9.20  
}  
snatpool other_snatpool {  
  member 199.5.6.30  
  member 207.8.9.30  
}
```

Figure 19.2 bigip.conf entries for three SNAT pools

Finally, using the Configuration utility, the ISP creates a SNAT that maps each original IP address directly to the appropriate SNAT pool. Figure 19.3 shows these mappings as they appear in the **bigip.conf** file.

```
snat map {  
  192.1.1.10 192.1.1.11 to snatpool customer1_snatpool  
}  
  
snat map {  
  192.1.1.20 192.1.1.21 to snatpool customer2_snatpool  
}  
  
snat map default to snatpool other_snatpool
```

Figure 19.3 bigip.conf entries that map original addresses to SNAT pools

Example 2 - Establishing an intelligent SNAT

If you want to base SNAT mapping on criteria other than the original client IP address, such as a server port, you can write an iRule and specify a SNAT pool within the iRule. In this case, you use the SNAT screens in the Configuration utility to create a SNAT pool only, and not an actual SNAT object.

For example, suppose a user such as an ISP has two redundant connections to the Internet. In addition, the ISP handles many simultaneous CHAT connections (using port **531**), and wants to avoid exhausting the supply of server-side client ports. Finally, the ISP wants to collect statistics separately for CHAT, SMTP, and all other traffic. In this case, configuring an intelligent SNAT is the best way to choose the translation address.

To implement the intelligent SNAT, the ISP takes the following steps.

First, the ISP creates a load balancing pool called **out_pool**. In the **bigip.conf** file, the pool looks like the sample in Figure 19.4.

```
pool out_pool {  
  lb_method round_robin  
  member 199.5.6.254:0  
  member 207.8.9.254:0  
}
```

Figure 19.4 *bigip.conf* entries for a pool to be used in an intelligent SNAT

Next, as shown in Figure 19.5, the ISP uses the Configuration utility to create a SNAT pool called **chat_snatpool** containing four IP addresses: **199.5.6.10**, **199.5.6.11**, **207.8.9.10**, and **207.8.9.11**. The BIG-IP system automatically designates these IP addresses as translation addresses during creation of the SNAT pool. These addresses correspond to each of the two next hop networks that are to be used for CHAT traffic. In the **bigip.conf** file, the SNAT pool looks like the sample in Figure 19.5.

```
snatpool chat_snatpool {  
  member 199.5.6.10  
  member 199.5.6.11  
  member 207.8.9.10  
  member 207.8.9.11  
}
```

Figure 19.5 A SNAT pool definition for CHAT traffic

Next, for each translation address, the ISP uses the Configuration utility to change the timeout value for TCP connections to **600**.

Then the ISP creates a second SNAT pool, **smtp_snatpool** containing two translation addresses: **199.5.6.20** and **207.8.9.20**. Each address corresponds to one of the two next hop networks that are to be used for SMTP traffic. In the **bigip.conf** file, the SNAT pool looks like the sample in Figure 19.6.

```
snatpool smtp_snatpool {  
  member 199.5.6.20  
  member 207.8.9.20  
}
```

Figure 19.6 A SNAT pool definition for SMTP traffic

Next, the ISP creates the SNAT pool **other_snatpool** for all other traffic (that is, non-CHAT and non-SMTP traffic), where each IP address corresponds to one of the two next hop networks that are to be used by all other traffic. This is shown in Figure 19.7.

```
snatpool other_snatpool { \SNAT pool definition
member 199.5.6.30
member 207.8.9.30
}
```

Figure 19.7 A SNAT pool definition for all other traffic

Then the ISP writes an iRule that selects both a SNAT pool, based on the server port of the initiating packet, and the load balancing pool **out_pool**. Figure 19.9, shows how the iRule specifies the command **TCP::local_port** to indicate the type of packet data to be used as a basis for selecting translation addresses. The iRule also shows the command **snatpool** (shown in Figure 19.8) to specify the SNAT pools from which the BIG-IP system is to select the translation addresses.

```
rule my_iRule {
when SERVER_CONNECTED
if ( TCP::local_port equals 531 ) {
use snatpool chat_snatpool
}
else if ( TCP::local_port equals 25 ) {
use snatpool smtp_snatpool
}
else {
use snatpool other_snatpool
}
use pool out_pool
}
```

Figure 19.8 Example of an iRule that references an intelligent SNAT

The **if** statement in the iRule instructs the BIG-IP system to test the value of server port specified in the header of the client request. Based on the results, the BIG-IP system selects both a SNAT pool and a load balancing pool.

As a final step, the ISP assigns the iRule as a resource to a wildcard virtual server, as shown in Figure 19.9.

```
virtual 0.0.0.0:0 use rule my_iRule
```

Figure 19.9 Assignment of an iRule to a wildcard virtual server



20

Configuring Nodes

- Introducing nodes
- Creating and modifying nodes
- Configuring node settings
- Managing nodes

Introducing nodes

Nodes are the network devices to which a BIG-IP® Link Controller™ system passes traffic. You can explicitly create a node, or you can instruct the BIG-IP system to automatically create one when you add a pool member to a load balancing pool.

The difference between a node and a pool member is that a node is designated by the device's IP address only (**10.10.10.10**), while designation of a pool member includes an IP address and a service (such as **10.10.10:80**).

A primary feature of nodes is their association with health monitors. Like pool members, nodes can be associated with health monitors as a way to determine server status. However, a health monitor for a pool member reports the status of a service running on the device, whereas a health monitor associated with a node reports status of the device itself.

For example, if an ICMP health monitor is associated with node **10.10.10.10**, which corresponds to pool member **10.10.10.10:80**, and the monitor reports the node as being in a **down** state, then the monitor also reports the pool member as being **down**. Conversely, if the monitor reports the node as being in an **up** state, then the monitor reports the pool member as being either **up** or **down**, depending on the status of the service running on it.

You create a node using the Configuration utility, and then adjust the settings as needed. Using the same utility, you can also display information about nodes, enable and disable nodes, and delete nodes.

Creating and modifying nodes

Nodes are the basis for creating a load balancing pool. For any server that you want to be part of a load balancing pool, you must first create a node, that is, designate that server as a node. After designating the server as node, you can add the node to a pool as a pool member. You can also associate a health monitor with the node, to report the status of that server. For information on adding nodes to load balancing pools, see Chapter 5, *Configuring Load Balancing Pools*.

◆ Note

If you create a pool member without first creating the corresponding node, the BIG-IP system automatically creates the node for you.

You use the Configuration utility to create a node. When you create a node, the BIG-IP system automatically assigns a group of default settings to that node. You can retain these default settings or modify them. You can also modify the settings at a later time, after you have created the node. For information on these settings, see either *Configuring node settings*, on page 20-3, or the online help.

It is helpful to understand that the BIG-IP system designates some settings as basic and others as advanced. If you decide to modify some of the default settings when you create the node, be sure to select the **Advanced** option on the screen to view all configurable settings. For more information on basic and advanced settings, see Chapter 1, *Introducing the Link Controller*.

To create a node

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. In the upper-right corner of the screen, click **Create**.
The New Node screen opens.
3. For the **Address** setting, type the IP address of the node.
4. Specify, retain, or change each of the other settings.
5. Click **Finished**.

To modify an existing node

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. In the Address column, click an address.
The settings for the node displays.
3. Retain or modify any node settings.
4. Click **Update**.

Configuring node settings

You can configure node settings to tailor nodes to your specific needs. For those settings that have default values, you can retain those default settings or modify them. Also, you can modify settings either when you create the node, or at any time after you have created it.

Table 20.1 lists these configurable settings and their default values. Following this table are descriptions of specific settings.

Node settings	Description	Default Value
Name	Specifies the name of the node.	No default value
Address	Specifies the IP address of the node. This setting is required.	No default value
Description	Specifies a unique description of the node	No default value
Health Monitors	Defines whether the BIG-IP system should associate the default monitor with the node, or whether you want to specifically assign a monitor to the node.	Node default
Select Monitors	Specifies monitors that you want to associate with the node. This setting is only available when you set the Health Monitors setting to Node Specific .	No default value
Availability Requirement	Specifies the minimum number of health monitors that must report a node as being available to receive traffic before the BIG-IP system reports that node as being in an up state. This setting is only available when you set the Health Monitors setting to Node Specific .	All
Ratio	Specifies the ratio weight you want to assign to the node.	1
Connection Limit	Specifies the maximum number of concurrent connections allowed on a node.	0

Table 20.1 Node configuration settings

Specifying an address for a node

For each node that you configure, you must specify an IP address. An example of a node IP address is **10.10.10.10**. This is the only required setting.

Specifying a node name

For each node that you configure, you can give it a unique node name, such as **Node_1**. Node names are case-sensitive and may contain letters, numbers, and underscores (_) only. Reserved keywords are not allowed.

Assigning health monitors

Using the BIG-IP system, you can monitor the health or performance of your nodes by associating monitors with those nodes. This is similar to associating a monitor with a load balancing pool, except that in the case of nodes, you are monitoring the IP address, whereas with pools, you are monitoring the services that are active on the pool members.

The BIG-IP system contains many different pre-configured monitors that you can associate with nodes, depending on the type of traffic you want to monitor. You can also create your own custom monitors and associate them with nodes. The only pre-configured monitors that are not available for associating with nodes are monitors that are specifically designed to monitor pools or pool members rather than nodes.

There are two ways that you can associate a monitor with a node: by assigning the same monitor (that is, a default monitor) to multiple nodes at the same time, or by explicitly associating a monitor with each node as you create it.

For more information about health and performance monitors, see Chapter 8, *Configuring Monitors*.

Specifying a default monitor

As explained earlier in this chapter, if you create a pool member without first creating the parent node, the BIG-IP system automatically creates the parent node for you. Fortunately, you can configure the BIG-IP system to automatically associate one or more monitor types with every node that the BIG-IP system creates. This eliminates the task of having to explicitly choose monitors for each node.

To associate one or more monitors with every node by default, you must first specify the monitors that you want to assign to nodes (see *To specify one or more default monitors*, following). Once you have performed this task, the BIG-IP system then assigns the specified default monitors to any node that the system automatically creates.

To specify one or more default monitors

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. On the menu bar, click **Default Monitor**.
3. For the **Health Monitors** setting, locate the **Available** box, and select a health monitor.
4. Click the Move button (<<) to move the monitor name to the **Active** box.
5. Repeat for each monitor that you want to designate as a default monitor.
6. From the **Configuration** list, select **Advanced**.
7. From the **Availability Requirement** list, select how to define a node as **up**:
 - Select **All**.
This specifies that all active monitors must succeed before the node is considered to be **up**.
 - Select **At Least** and then type a number.
This specifies that the designated number of monitors must succeed before the node is considered to be **up**.
8. Click **Update**.

Explicitly associating monitors with a node

Sometimes, you might want to explicitly create a node, rather than having the BIG-IP system create the node automatically. In this case, when you create the node and configure its **Health Monitors** setting, you can either:

- ◆ **Associate other monitors with the node**
To associate other monitors (that is, non-default monitors) with a node, you set the value of the node's **Health Monitors** setting to **Node Specific**, when you create the node or modify the node's settings. The Configuration utility then allows you to choose from a list of monitors that are available for associating with that node.
- ◆ **Associate the default monitors with the node**
To associate the default monitors with a node, you set the value of the **Health Monitors** setting to **Node Default**.

Specifying the availability requirement

By configuring the **Availability Requirement** setting, you can specify the minimum number of health monitors that must report a node as being available to receive traffic before the BIG-IP system reports that node as being in an **up** state. Acceptable values are **All**, or a number that you specify. If you choose the value **At Least**, you then specify a number.

Specifying a ratio weight

The **Ratio** setting specifies a ratio weight for the node. The default setting is **1**. For information on ratio weights, see Chapter 5, *Configuring Load Balancing Pools*.

Setting a connection limit

Using the **Connection Limit** setting, you can specify the maximum number of concurrent connections allowed for a node. Note that the default value of **0** (zero) means that there is no limit to the number of concurrent connections that the node can receive.

Managing nodes

After you have created your nodes and configured their settings to suit your needs, you might want to perform some additional management tasks. Using the Configuration utility, you can:

- View a list of nodes
- View node properties
- Display and understand node status
- Enable or disable existing nodes
- Delete existing nodes
- Disable monitor associations

Viewing a list of nodes

You can view a list of the existing nodes that you have permission to view. When you display the list of nodes, the Configuration utility displays the following information about each node:

- Status
- Node address
- Node name

Use the following procedure to view a list of nodes defined on the BIG-IP system.

To view the list of nodes

On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.

Viewing node properties

You can use the Configuration utility to view the general properties of a node. These properties and their descriptions are:

- ◆ **Address**
The IP address of the node.
- ◆ **Availability**
The status of the node.
- ◆ **Health monitors**
The health monitors that are associated with the node.
- ◆ **Current connections**
The number of current connections that the node has received.

◆ **State**

The state of the traffic that you want the node to receive. Possible states are:

- **Enabled (All Traffic Allowed)**
- **Disabled (Only persistent or active connections allowed)**
- **Forced offline (Only active connections allowed)**

To view node properties

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. In the Address column, click the address of the node you want to view.
The settings for the node display.

Understanding node status

At any time, you can determine the status of a node using the Configuration utility. You can find this information by displaying the list of nodes and viewing the Status column, or by viewing the **Availability** property of a node.

The Configuration utility indicates status by displaying one of several icons, distinguished by shape and color:

- The shape of the icon indicates the status that the monitor has reported for that node.
- The color of the icon indicates the actual status of the node.

To understand these icons with respect to node status, see Table 20.2, on page 20-8. To display the icons within the Configuration utility, see *To view node properties* on this page.



Status indicator	Explanation
	The node is enabled and able to receive traffic.
	The node is enabled but is currently unavailable. However, the node might become available later, with no user action required. An example of an unavailable node becoming available automatically is when the number of concurrent connections to the node no longer exceeds the value defined in the node's Connection Limit setting.

Table 20.2 Explanation of status icons for nodes






Status indicator	Explanation
	The node is enabled but offline because an associated monitor has marked the node as down . To change the status so that the node can receive traffic, user intervention is required.
	The node is set to Disabled , although a monitor has marked the node as up .
	The node is set to Disabled and is down .
	The node is set to Disabled and is offline either because a user disabled it, or a monitor has marked the node as down .
	The status of the node is unknown. Sample reasons for unknown node status are: The node has no monitor associated with it. Monitor results are not available yet. The node's IP address is misconfigured. The node has been disconnected from the network.

Table 20.2 Explanation of status icons for nodes

Enabling or disabling a node

A node must be enabled in order to accept traffic. When a node is disabled, the BIG-IP system allows existing connections to time out or end normally. In this case, the node can accept new connections only if the connections belong to an existing persistence session. (In this way a disabled node differs from a node that is set to **down**. The **down** node allows existing connections to time out, but accepts no new connections.)

To enable or disable a node

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. Locate the address of the node you want to enable or disable, and in the column to the left, check the Select box.
3. At the bottom of the screen, click **Enable** or **Disable**.

Deleting a node

If you are no longer using a node in a pool, you can delete the node.

To delete a node

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. Locate the address of the node you want to enable or disable, and in the column to the left, check the Select box.
3. On the bottom of the screen, click **Delete**.
The Delete confirmation screen opens.
4. Click **Delete**.

Removing monitor associations

Using the Configuration utility, you can remove a monitor that is explicitly associated with a specific node. When removing a monitor associated with a specific node, you can either remove the monitor association altogether, or change it so that only the default monitor is associated with the node.

Alternatively, you can remove any default monitors, that is, monitors that the BIG-IP system automatically associates with any node that you create.

For more information on monitor associations, see *Assigning health monitors*, on page 20-4.

To remove an explicit monitor association for a node

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. Click the address of the node you want to manage.
3. In the Configuration area, from the **Health Monitors** list, select **Node Default** or **None**.
4. Click **Update**.

To remove a default monitor

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
The Node List screen opens.
2. On the menu bar, click **Default Monitor**.
3. Using the Move button (>>), move any active monitors from the **Active** box to the **Available** box.
4. Click **Update**.



21

Configuring Rate Shaping

- Introducing rate shaping
- Creating and implementing rate classes
- Configuring rate class settings
- Managing rate classes

Introducing rate shaping

The BIG-IP® Link Controller™ system includes a feature called rate shaping. **Rate shaping** allows you to enforce a throughput policy on incoming traffic. Throughput policies are useful for prioritizing and restricting bandwidth on selected traffic patterns.

Rate shaping can be useful for an e-commerce site that has preferred clients. For example, the site might want to offer higher throughput for preferred customers, and lower throughput for other site traffic.

The rate shaping feature works by first queuing selected packets under a rate class, and then dequeuing the packets at the indicated rate and in the indicated order specified by the rate class. A **rate class** is a rate-shaping policy that defines throughput limitations and a packet scheduling method to be applied to all traffic handled by the rate class.

You configure rate shaping by creating one or more rate classes and then assigning the rate class to a packet filter or to a virtual server. You can also use the iRules® feature to instruct the BIG-IP system to apply a rate class to a particular connection.

You can apply a rate class specifically to traffic from a server to a client or from a client to a server. If you configure the rate class for traffic that is going to a client, the BIG-IP system does not apply the throughput policy to traffic destined for the server. Conversely, if you configure the rate class for traffic that is going to a server, the BIG-IP system does not apply the throughput policy to traffic destined for the client.

To configure rate shaping, you use the Rate Shaping screens within the Network section of the Configuration utility.

Creating and implementing rate classes

A **rate class** defines the throughput limitations and packet scheduling method that you want the BIG-IP system to apply to all traffic that the rate class handles. You assign rate classes to virtual servers and packet filter rules, as well as through iRules.

If the same traffic is subject to rate classes that you have assigned from more than one location, the BIG-IP system applies the last-assigned rate class only. The BIG-IP system applies rate classes in the following order:

- The first rate class that the BIG-IP system assigns is from the last packet filter rule that matched the traffic and specified a rate class.
- The next rate class that the BIG-IP system assigns is from the virtual server; if the virtual server specifies a rate class, the rate class overrides any rate class that the packet filter selects.
- The last rate class assigned is from the iRule; if the iRule specifies a rate class, this rate class overrides any previously-selected rate class.

To create a rate class

1. On the Main tab, expand **Network**, and click **Rate Shaping**. The Rate Class List screen opens.
2. In the upper-right corner of the screen, click **Create**.
3. Specify whether you want to enable the rate class to borrow bandwidth from a parent rate class:
 - If you do not want the rate class to borrow bandwidth from a parent class, select **Basic**. For more information, see *Borrowing bandwidth*, on page 21-7.
 - If you want to enable the rate class to borrow bandwidth from a parent class, select **Advanced**. For more information, see *Specifying a parent class*, on page 21-7.
4. Configure all settings as needed.
For information on settings, see *Configuring rate class settings*, on page 21-3, or see the online help.
5. Click **Finished**.

After you have created a rate class, you must assign it to a virtual server or a packet filter rule, or you must specify the rate class from within an iRule.

- For more information on virtual servers, see Chapter 6, *Configuring Virtual Servers*.
- For more information on packet filter rules, access the Packet Filters screens within the Configuration utility and display the online help.
- For more information on iRules, see Chapter 18, *Writing iRules*.

Configuring rate class settings

When you create a rate class, the BIG-IP system assigns some default settings to the rate class. You can retain these default settings or modify them to suit your needs. The settings that you can configure for a rate class are described in Table 21.1.

Setting	Description	Default Value
Name	Specifies a unique name for the rate class. Every rate class requires a name.	No default value
Base Rate	Specifies the base throughput rate allowed for traffic that the rate class handles. Packets are generally not allowed to exceed the specified rate. This setting is required.	No default value
Ceiling Rate	Similar to the base rate, but specifies a hard, absolute limit. This number specifies the absolute limit on the rate at which traffic is allowed to flow when bursting or borrowing. For information on bandwidth bursting and borrowing, see <i>Specifying a burst size</i> , on page 21-4.	Same as Base Rate
Burst Size	Specifies the maximum number of bytes that traffic is allowed to burst beyond the base rate, before needing to borrow bandwidth. When this value is set to 0 , no bursting is allowed. For information on bandwidth bursting and borrowing, see <i>Specifying a burst size</i> , on page 21-4.	0
Direction	Specifies the direction of traffic to which the rate class is applied. Possible values are Any , Client , and Server .	Any
Parent Class	Specifies the rate class from which this class can borrow bandwidth. A child rate class can borrow any unused bandwidth from the parent rate class, thereby supplementing the burst size of the child rate class. This is an Advanced setting. For information on bandwidth bursting and borrowing, see <i>Specifying a burst size</i> , on page 21-4.	None
Shaping Policy	Specifies a shaping policy that includes customized values for the Drop Policy and Queue Method settings.	None
Queue Method	Specifies the method that the rate class uses to queue and dequeue traffic. Allowed settings are sfq and pfifo .	Same as parent class if a parent class is specified; otherwise, Stochastic Fair Queue
Drop Policy	Specifies when and how to drop packets, if required, when the traffic handling queue is full. The possible values are: fred : Specifies that the system drops packets according to the type of traffic in the flow. red : Specifies that the system randomly drops packets. tail : Specifies that the system drops the end of the traffic stream.	tail

Table 21.1 Settings for configuring a rate class

Specifying a name

The first setting you configure for a rate class is the rate class name. Rate class names are case-sensitive and may contain letters, numbers, and underscores (_) only. Reserved keywords are not allowed.

Each rate class that you define must have a unique name. This setting is required.

To specify a rate class name, locate the **Name** box on the New Rate Class screen and type a unique name for the rate class.

Specifying a base rate

The **Base Rate** setting specifies the base throughput rate allowed for traffic that the rate class handles. Packets are generally not allowed to exceed the specified rate. You can specify the base rate in bits per second (bps), kilobits per second (Kbps), megabits per second (Mbps), or gigabits per second (Gbps). The default unit is bits per second. This setting is required.

The minimum base rate that you can configure is **296** bps.

◆ **Note**

These numbers are powers of 10, not powers of 2.

Specifying a ceiling rate

The **Ceiling Rate** setting specifies the absolute limit at which traffic is allowed to flow when bursting or borrowing. You can specify the ceiling rate in bits per second (bps), kilobits per second (Kbps), megabits per second (Mbps), or gigabits per second (Gbps). The default unit is bits per second.

If you specify a ceiling rate, the rate must be equal to or greater than the base rate. If you omit the ceiling rate or set it equal to the base rate, traffic throughput can never exceed the base rate.

Specifying a burst size

You use the **Burst Size** setting when you want to allow the rate of traffic flow that a rate class controls to exceed the base rate. Exceeding the base rate is known as *bursting*. When you configure a rate class to allow bursting (by specifying a value other than 0), the BIG-IP system saves any unused bandwidth and uses that bandwidth later to enable the rate of traffic flow to temporarily exceed the base rate. Specifying a burst size is useful for smoothing out traffic patterns that tend to fluctuate or exceed the base rate, such as HTTP traffic.

The value of the **Burst Size** setting defines the maximum number of bytes that you want to allow for bursting. Thus, if you set the burst size to 5,000 bytes, and the rate of traffic flow exceeds the base rate by 1,000 bytes per second, then the BIG-IP system allows the traffic to burst for a maximum of five seconds.

When you specify a burst size, the BIG-IP system creates a burst reservoir of that size. A *burst reservoir* stores bandwidth that the BIG-IP system uses for bursting later. The burst reservoir becomes depleted as the rate of traffic flow exceeds the base rate, and is replenished as the rate of traffic falls below the base rate. The **Burst Size** value that you configure in a rate class thus represents:

- The maximum number of bytes that the rate class is allowed to transmit when the traffic-flow rate exceeds the base rate
- The maximum number of bytes that the BIG-IP system can replenish into the burst reservoir
- The amount of bandwidth initially available for bursting beyond the base rate

The burst size is measured in bytes. For example, a value of either **10000** or **10K** equals 10,000 bytes. The default value is **0**.

Depleting the burst reservoir

When the rate of traffic flow exceeds the base rate, the BIG-IP system automatically depletes the burst reservoir, at a rate determined by the number of bytes per second that the traffic flow exceeds the base rate.

Continuing with our previous example in which traffic flow exceeds the base rate by 1,000 bytes per second, if the traffic-flow rate only exceeds the base rate for two seconds, then 2,000 bytes are depleted from the burst size and the maximum bytes available for bursting decreases to 3,000.

Replenishing the burst reservoir

When the rate of traffic flow falls below the base rate, the BIG-IP system stores the unused bandwidth (that is, the difference between the base rate and the actual traffic-flow rate) in the burst reservoir. Later, the BIG-IP system uses this bandwidth when traffic flow exceeds the base rate. Thus, the BIG-IP system replenishes the burst reservoir whenever it becomes depleted due to traffic flow exceeding the base rate.

The size of the burst reservoir cannot exceed the specified burst size. For this reason, the BIG-IP system replenishes the reservoir with unused bandwidth only until the reservoir reaches the amount specified by the

Burst Size setting. Thus, if the burst size is set to **5,000**, then the BIG-IP system can store only 5,000 bytes of unused bandwidth for later use when the rate of traffic flow exceeds the base rate.

◆ **Note**

Specifying a burst size does not allow the rate class to exceed its ceiling rate.

Specifying a non-zero burst size

The following example illustrates the behavior of the BIG-IP system when you set the **Burst Size** setting to a value other than **0**.

This example shows throughput rates in units of bytes-per-second instead of the default bits-per-second. This is only to simplify the example. You can derive bytes-per-second from bits-per-second by dividing the bits-per-second amount by 8.

Suppose you configure the rate class settings with these values:

- Base rate: 1,000 bytes per second
- Ceiling rate: 4,000 bytes per second
- Burst size: 5,000 bytes

Consider the following scenario:

◆ **If traffic is currently flowing at 800 bytes per second**

No bursting is necessary because the rate of traffic flow is below the base rate defined in the rate class.

Because the traffic is flowing at 200 bytes per second less than the base rate, the BIG-IP system can potentially add 200 bytes of unused bandwidth to the burst reservoir. However, because no bursting has occurred yet, the reservoir is already full at the specified 5,000 bytes, thus preventing the BIG-IP system from storing the 200 bytes of unused bandwidth in the reservoir. In this case, the BIG-IP system simply discards the unused bandwidth.

◆ **If traffic climbs to 1,000 bytes per second (equal to the base rate)**

Still no bursting occurs, and there is no unused bandwidth.

◆ **If traffic jumps to 2,500 bytes per second**

For each second that the traffic continues to flow at 2,500 bytes per second, the BIG-IP system empties 1,500 bytes from the burst reservoir (the difference between the traffic flow rate and the base rate). This allows just over three seconds of bursting at this rate before the burst reservoir of 5,000 bytes is depleted. Once the reservoir is depleted, the BIG-IP system reduces the traffic flow rate to the base rate of 1,000 bytes per second, with no bursting allowed.

◆ **If traffic drops back down to 800 bytes per second**

No bursting is necessary, but now the BIG-IP system can add the 200 bytes per second of unused bandwidth back into the burst reservoir because the reservoir is empty. If traffic continues to flow at 800 bytes

per second, the burst reservoir becomes fully replenished from 0 to 5,000 bytes in 25 seconds (at a rate of 200 bytes per second). If traffic stops flowing altogether, creating 1,000 bytes per second of unused bandwidth, then the BIG-IP system adds 1,000 bytes per second into the burst reservoir, thus replenishing the reservoir from 0 to 5,000 bytes in only 5 seconds.

Borrowing bandwidth

In some cases, a rate class can borrow bandwidth from the burst reservoir of its parent class. For more information, see *Specifying a parent class*, following.

Specifying direction

Using the **Direction** setting, you can apply a rate class to client or server traffic. Thus, you can apply a rate class to traffic going to a client, to a server, or to both client and server. Possible values are **Any**, **Client**, and **Server**. The default value is **Any**.

Specifying direction is useful in cases where the nature of the traffic is directionally-biased. For example, if you offer an FTP service to external clients, you might be more interested in limiting throughput for those clients uploading files to your site than you are for clients downloading files from your site. In this case, you would select **Server** as the direction for your FTP rate class, because the **Server** value only applies your throughput restriction to traffic going from the client to the server.

Specifying a parent class

When you create a rate class, you can use the **Parent Class** setting to specify that the rate class has a parent class. This allows the rate class to borrow unused bandwidth from that parent class. A child class can borrow unused bandwidth from its parent, but a parent class cannot borrow from a child class. Borrowing is also not possible between two child classes of the same parent class or between two unrelated rate classes.

You specify a parent class by displaying the New Rate Class screen and selecting **Advanced**, and then selecting a rate class name in the **Parent Class** setting.

A parent class can itself have a parent, provided that you do not create a circular dependency. A *circular dependency* is a relationship where a rate class is a child of itself, directly or indirectly.

If a rate class has a parent class, the child class can take unused bandwidth from the parent class. The process occurs in this way:

- If the rate of traffic flow to which the child class is applied exceeds its base rate, the child class begins to deplete its burst reservoir as described previously.

- If the reservoir is empty (or no burst size is defined for the rate class), then the BIG-IP system takes unused base-rate bandwidth from the parent class and gives it to the child class.
- If the unused bandwidth from the parent class is depleted, then the child class begins to use the reservoir of the parent class.
- If the reservoir of the parent class is empty (or no burst size is defined for the parent class), then the child class attempts to borrow bandwidth from the parent of the parent class, if the parent class has a parent class.
- This process continues until there is no remaining bandwidth to borrow or there is no parent from which to borrow.

Borrowing only allows the child to extend its burst duration; the child class cannot exceed the ceiling rate under any circumstance.

◆ **Note**

Although the above description uses the term "borrowing," bandwidth that a child class borrows is not paid back to the parent class later, nor is unused bandwidth of a child class returned to its parent class.

Specifying a queue method

The **Queue Method** setting determines the method and order in which the BIG-IP system dequeues packets.

A rate class supports two queue disciplines:

◆ **sfq**

Stochastic Fair Queueing (SFQ) is a queueing method that queues traffic under a set of many lists, choosing the specific list based on a periodically-changing hash of the connection information. This results in traffic from the same connection always being queued in the same list. SFQ then dequeues traffic from the set of the lists in a round-robin fashion. The overall effect is that fairness of dequeuing is achieved because one high-speed connection cannot monopolize the queue at the expense of slower connections.

◆ **pfifo**

The **Priority FIFO (PFIFO)** queueing method queues all traffic under a set of five lists based on the Type of Service (ToS) field of the traffic. Four of the lists correspond to the four possible ToS values (**Minimum delay**, **Maximum throughput**, **Maximum reliability**, and **Minimum cost**). The fifth list represents traffic with no ToS value. The PFIFO method then processes these five lists in a way that attempts to preserve the meaning of the ToS field as much as possible. For example, a packet with the ToS field set to **Minimum cost** might yield dequeuing to a packet with the ToS field set to **Minimum delay**.

Managing rate classes

Once you have created a rate class, you can use the Configuration utility to list existing rate classes, view or modify the settings of a rate class, or delete a rate class.

To list existing rate classes

1. On the Main tab, expand **Network**, and click **Rate Shaping**.
The Rate Class List screen opens.
2. View the list of rate classes.

To view or modify a rate class

1. On the Main tab, expand **Network**, and click **Rate Shaping**.
The Rate Class List screen opens.
2. Click a rate class name in the list.
The settings for the rate class display.
3. Retain or modify any setting values. For information rate class settings, see *Configuring rate class settings*, on page 21-3.
4. Click **Update**.

To delete a rate class

1. On the Main tab, expand **Network**, and click **Rate Shaping**.
The Rate Class List screen opens.
2. Locate a rate class name in the list, and to the left of the name, check the Select box.
3. At the bottom of the screen, click **Delete**.
The Delete confirmation screen displays.
4. Click **Delete**.
The rate class is deleted.



A

Additional Monitor Considerations

- Implementing monitors for Dynamic Ratio load balancing
- Implementing an MSSQL monitor

Implementing monitors for Dynamic Ratio load balancing

You can configure Dynamic Ratio load balancing for pools that consist of RealNetworks® RealServer servers, Windows® servers equipped with Windows Management Instrumentation (WMI), or any server equipped with an SNMP agent such as the UC Davis SNMP agent or Windows® 2000 Server SNMP agent.

To implement Dynamic Ratio load balancing for these types of servers, the BIG-IP® Link Controller™ system provides a special monitor plug-in file and a health or performance monitor for several types of servers. The exception is a server equipped with an SNMP agent. In this case, the BIG-IP system provides the monitor only; no special plug-in file is required for a server running an SNMP agent.

You must install the monitor plug-in on each server to be monitored, and you must create a performance monitor that resides on the BIG-IP system. Once you have created a monitor, the monitor communicates directly with the server plug-in. For each server type, Table A.1 shows the required monitor plug-in and the corresponding performance monitor types.

Server Type	Monitor plug-in	Monitor Type
RealServer Windows server	F5RealMon.dll	Real Server
RealServer UNIX server	f5realmon.so	Real Server
Windows server with WMI	F5lsapi.dll	WMI
Windows 2000 Server server	SNMP agent	SNMP DCA and SNMP DCA Base
UNIX server	UC Davis SNMP agent	SNMP DCA and SNMP DCA Base

Table A.1 Monitor plug-ins and corresponding monitor templates

Implementing a Real Server monitor

For RealSystem Server systems, the BIG-IP system provides a monitor plug-in that gathers the necessary metrics when you have installed the plug-in on the RealSystem Server system. Configuring a RealSystem Server for Dynamic Ratio load balancing consists of four tasks:

- Installing the monitor plug-in on the RealSystem server
- Configuring a Real Server monitor on the BIG-IP system
- Associating the monitor with the server to gather the metrics
- Creating or modifying the server pool to use Dynamic Ratio load balancing

To install the monitor plug-in on a RealSystem Server system (Windows version)

1. Download the monitor plug-in **F5RealServerPlugin.dll** from the BIG-IP system.
The plug-in is located in the folder **/usr/local/www/docs/agents**.
2. Copy **F5RealServerPlugin.dll** to the RealServer **plug-ins** directory.
(For example, **C:\Program Files\RealServer\plug-ins**.)
3. If the RealSystem Server process is running, restart it.

To install and compile a Linux or UNIX RealSystem Server monitor plug-in

1. Using the **.iso** image, burn a CD-ROM of the BIG-IP system software.
2. On the CD, navigate to the directory **/downloads/rsplug-ins**.
3. Copy the file **F5RealMon.src.tar.gz** to the directory **/var/tmp** on the BIG-IP system.
4. On the BIG-IP system, change to the directory **/var/tmp**:

```
cd /var/tmp
```
5. Use the UNIX **tar** command to uncompress the file **F5RealMon.src.tar.gz**:

```
tar -xvzf F5RealMon.src.tar
```
6. Change to the **F5RealMon.src** directory:

```
cd F5RealMon.src
```
7. Type the **ls** command to view the directory contents.
8. To compile the source, use the instructions in the file **build_unix_note**.
9. Start RealSystem Server.

Once the plug-in is installed and compiled, you must configure a Real Server monitor, associate the configured monitor with the pool member (a RealSystem Server server), and set the load balancing method to Dynamic Ratio:

- To configure a Real Server monitor, see Chapter 8, *Configuring Monitors*.
- To associate the performance monitor with the pool member, see Chapter 5, *Configuring Load Balancing Pools*.
- To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 5, *Configuring Load Balancing Pools*.

Implementing a WMI monitor

For Windows system running Windows Management Instrumentation (WMI), the BIG-IP system provides a Data Gathering Agent **F5Isapi.dll** for the server. Configuring a Windows platform for Dynamic Ratio load balancing consists of four tasks:

- Installing the Data Gathering Agent **F5Isapi.dll** on the server
- Configuring a WMI monitor on the BIG-IP system
- Associating the monitor with the server to gather the metrics
- Creating or modifying the server pool to use the Dynamic Ratio load balancing method

The procedure for installing the Data Gathering Agent on a server differs according to whether the server is running Internet Information Services (IIS) for Windows[®] Server version 5.0 or IIS version 6.0.

To install the Data Gathering Agent (F5Isapi) on an IIS 5.0 server

1. Download the **Data Gathering Agent (F5Isapi.dll)** from the BIG-IP system.
You can find this plug-in in either the **/var/windlls** or the **/usr/local/www/docs/agents** directory on the BIG-IP system.
2. Copy **f5isapi.dll** to the directory **C:\inetpub\scripts**.
3. Open the Internet Services Manager.
4. In the left pane of the Internet Services Manager, open the folder **<machine_name>\Default Web Site\Script**, where **<machine_name>** is the name of the server you are configuring. The contents of **Scripts** folder opens in the right pane.
5. In the right pane, right click **F5Isapi.dll**, and select **Properties**. The Properties dialog box for **F5Isapi.dll** opens.
6. Deselect **Logvisits**.
(Logging of each visit to the agent quickly fills up the log files.)
7. Click the File Security tab.
The File Security options displays.
8. In the **Anonymous access and authentication control group** box, click **Edit**.
The Authentication Methods dialog box opens.
9. In the dialog box, clear all check boxes, then select **Basic Authentication**.
10. In the **Authentication methods** dialog box, click **OK** to accept the changes.
11. In the **Properties** dialog box, click **Apply**.
The WMI Data Gathering Agent is now ready to be used.

To install the Data Gathering Agent (F5Isapi) on an IIS 6.0 server

1. Create a **scripts** directory under the web site document root (**C:\InetPub\wwwroot** for **Default Website**).
2. Set the properties of the **scripts** directory to **scripts and executables**.
3. Copy the file **f5isapi.dll** to the created **scripts** directory.
4. Start IIS manager (**inetmgr**) and navigate to the **scripts** directory.
5. On the right pane, select the file name **f5isapi.dll**.
6. Select **Properties->File Security->Authentication and Access Control** and ensure that the settings **anonymous user** and **Basic Authentication** are checked.
7. If you want to allow all unknown extensions, then in IIS Manager, navigate to **Web Server Extensions -> All Unknown ISAPI extensions** and allow all unknown extensions. Otherwise, proceed to step 8.
8. If you want to allow the file **f5isapi.dll** only, navigate to **Web Server Extensions -> Tasks: Add a New Webserver Extension**. Then:
 - a) In the **Name** field, select **F5 ISAPI** and click **Add** for the required files.
This requests a path to the file.
 - b) Browse to the file **f5isapi.dll**, using the path **C:\InetPub\wwwroot\scripts\f5isapi.dll** for **Default Website**, and click **OK**.
 - c) Check the **Set Extension Status to Allowed** box, and click **OK**.
The value **F5 ISAPI** should now appear in the extensions list as **Allowed**.

Once you have installed the plug-in, you must configure a WMI monitor, associate the configured monitor with the pool member, and set the load balancing to Dynamic Ratio:

- To configure a WMI monitor, see Chapter 8, *Configuring Monitors*.
- To associate the custom monitor with the pool member, see Chapter 5, *Configuring Load Balancing Pools*.
- To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 5, *Configuring Load Balancing Pools*.

Implementing an SNMP DCA or SNMP DCA Base monitor

The BIG-IP system includes an SNMP data collecting agent that can query remote SNMP agents of various types, including the UC Davis agent and the Windows 2000 Server agent.

The BIG-IP system provides two monitor types that you can use to create a performance monitor for a server that uses an SNMP agent. These two monitor types are:

- ◆ **SNMP DCA**

Use this monitor when you want to use default values or specify new values for CPU, memory, and disk metrics. When using this template, you can also specify values for other types of metrics that you wish to gather.

- ◆ **SBMP DCA Base**

Use this monitor when you want to use default values or specify values for metrics other than CPU, memory, and disk usage. When using this monitor, values for CPU, memory, and disk metrics are omitted.

Configuring a server to use its SNMP agent for Dynamic Ratio load balancing consists of three tasks: configuring an SNMP DCA or SNMP DCA Base monitor, associating the monitor with the applicable pool member, and setting the load balancing method on the pool to the Dynamic Ratio method. For more information, see the following chapters or sections of this guide:

- To configure an SNMP DCA or SNMP DCA Base monitor, see Chapter 8, *Configuring Monitors*.
- To associate a monitor with the pool, see Chapter 5, *Configuring Load Balancing Pools*.
- To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 5, *Configuring Load Balancing Pools*.

Implementing an MSSQL monitor

Before you can use an MSSQL type of monitor, you must download a set of JDBC Java™ Archive (JAR) files from the Microsoft web site and install them on the BIG-IP system.

To download and install Microsoft JDBC files

1. Using an Internet browser, go to **www.microsoft.com**.
2. On the left panel, under Resources, click **Download**.
3. Select **SQL Server**.
4. In the **Keyword** field, type **JDBC Driver**.
5. Click **Go**.
A list of options displays.
6. Click **Microsoft SQL Server 2000 Driver for JDBC**.
7. Verify that you are downloading the UNIX **.tar** file, and not a Windows package. If not, go back to the previous screen.
8. On the right side of the screen, click **Download**.
9. When queried, select **Save the file to disk**.
10. Create a Linux directory, move the **.tar** file to that directory, and uncompress the file, using this command:

```
tar -xvf mssqlserver.tar
```


This command extracts four files: **EULA.txt**, **install.ksh**, **msjdbc.tar**, and **read.me**.
11. Install these files by typing **install.ksh** at the command line prompt. This untars the **msjdbc.tar** file, which creates several subdirectories.
12. Locate the **lib** subdirectory.
This directory contains three JAR files.
13. Copy the three JAR files to the BIG-IP system directory **/usr/bin/monitors/builtins/**.
You can now recursively remove the LINUX directory that you created in step 10.

After you install the JAR files, we recommend that you either reboot the BIG-IP system, or run the following command:

```
/usr/bin/monitors/builtins/DB_monitor cmd quit
```

Rebooting the system or running this command causes the BIG-IP system to recognize the newly-installed JAR files the next time that you run an MSSQL monitor.



Glossary

authentication

Authentication is the process of verifying a user's identity when the user is attempting to log on to a system.

bursting

Bursting is an aspect of rate shaping and occurs when the rate of traffic flow exceeds the base rate defined.

certificate

A certificate is an online credential signed by a trusted certificate authority and used for SSL network traffic as a method of authentication.

certificate authority (CA)

A certificate authority is an external, trusted organization that issues a signed digital certificate to a requesting computer system for use as a credential to obtain authentication for SSL network traffic.

chain

A chain is a series of filtering criteria used to restrict access to an IP address. The order of the criteria in the chain determines how the filter is applied, from the general criteria first, to the more detailed criteria at the end of the chain.

cipher

A cipher is an encryption/decryption algorithm that computer systems use when transmitting data using the SSL protocol.

configuration object

A configuration object is a user-created object that the Link Controller uses to implement a PAM authentication module. There is one type of configuration object for each type of authentication module that you create. See also *PAM (Pluggable Authentication Module)*.

Configuration utility

The Configuration utility is the browser-based application that you use to configure the BIG-IP system.

connection pooling

Connection pooling is an optimization feature that pools server-side connections for re-use by other client requests. Connection pooling reduces the number of new connections that must be opened for server-side client requests.

content switching

Content switching is the ability to load balance traffic based on data contained within a packet

custom profile

A custom profile is a profile that you create. A custom profile can inherit its default settings from a parent profile that you specify. See also *parent profile*.

default profile

A default profile is a profile that the Link Controller supplies with default setting values. You can use a default profile as is, or you can modify it. You can also specify it as a parent profile when you create a custom profile. You cannot create or delete a default profile. See also *profile*, *custom profile*.

default wildcard virtual server

A default wildcard virtual server has an IP address and port number of **0.0.0.0:0**, or ***:*** or **"any":"any"**. This virtual server accepts all traffic that does not match any other virtual server defined in the configuration.

destination address affinity persistence

Also known as sticky persistence, destination address affinity persistence supports TCP and UDP protocols, and directs session requests to the same server based solely on the destination IP address of a packet.

domain name

A domain name is a unique name that is associated with one or more IP addresses. Domain names are used in URLs to identify particular Web pages. For example, in the URL **http://www.siterequest.com/index.html**, the domain name is **siterequest.com**.

Dynamic Ratio load balancing method

Dynamic Ratio mode is like Ratio mode (see *Ratio method*), except that ratio weights are based on continuous monitoring of the servers and are therefore continually changing. Dynamic Ratio load balancing can be implemented on RealNetworks® RealServer platforms, on Microsoft® Windows® platforms equipped with Windows Management Instrumentation (WMI), or on a server equipped with either the UC Davis SNMP agent or Windows 2000 Server SNMP agent.

EAV (Extended Application Verification)

EAV is a health check that verifies an application on a node by running that application remotely. EAV health check is only one of the three types of health checks available on a Link Controller. See also *health check*, *health monitor*, and *external monitor*.

ECV (Extended Content Verification)

ECV is a health check that allows you to determine if a node is **up** or **down** based on whether the node returns specific content. ECV health check is only one of the three types of health checks available on a Link Controller. See also *health check*.

external monitor

An external monitor is a user-supplied health monitor. See also *health check*, *health monitor*.

external VLAN

The external VLAN is a default VLAN on the BIG-IP system. In a basic configuration, this VLAN has the administration ports locked down. In a normal configuration, this is typically a VLAN on which external clients request connections to internal servers. See also *VLAN*.

Fastest method

Fastest mode is a load balancing method that passes a new connection based on the fastest response of all currently active nodes.

forwarding virtual server

A forwarding virtual server is a virtual server that has no pool members to load balance. The virtual server simply forwards the packet directly to the destination IP address specified in the client request. See also *virtual server*.

health check

A health check is a Link Controller feature that determines whether a node is **up** or **down**. Health checks are implemented through health monitors. See also *health monitor*, *EAV (Extended Application Verification)*, *ECV (Extended Content Verification)*, *external monitor*.

health monitor

A health monitor checks a node to see if it is **up** and functioning for a given service. If the node fails the check, it is marked **down**. Different monitors exist for checking different services. See also *health check*, *EAV (Extended Application Verification)*, *ECV (Extended Content Verification)*, *external monitor*.

host virtual server

A host virtual server is a virtual server that represents a specific site, such as an Internet web site or an FTP site, and it load balances traffic targeted to content servers that are members of a pool.

HTTP redirect

An HTTP redirect sends an HTTP 302 Object Found message to clients. You can configure a pool with an HTTP redirect to send clients to another node or virtual server if the members of the pool are marked **down**.

ICMP (Internet Control Message Protocol)

ICMP is an Internet communications protocol used to determine information about routes to destination addresses.

intelligent SNAT

An intelligent SNAT is the mapping of one or more original client IP addresses to a translation address from within an iRule. Before writing an iRule to create an intelligent SNAT, you must create a SNAT pool. See also *SNAT pool*.

interface

The physical port on a BIG-IP system is called an interface.

internal VLAN

The internal VLAN is a default VLAN on the BIG-IP system. In a basic configuration, this VLAN has the administration ports open. In a normal configuration, this is a network interface that handles connections from internal servers.

IPsec

IPsec (Internet Protocol Security) is a communications protocol that provides security for the network layer of the Internet without imposing requirements on applications running above it.

iRule

An iRule is a user-written script that controls the behavior of a connection passing through the Link Controller. iRules® are an F5 Networks feature and are frequently used to direct certain connections to a non-default load balancing pool. However, iRules can perform other tasks, such as implementing secure network address translation and enabling session persistence.

JAR file

A JAR file is a file in Java™ Archive (JAR) file format that enables you to bundle multiple files into a single archive file. Typically, a JAR file contains the class files and auxiliary resources associated with applets and applications.

JDBC

JDBC is a Java™ technology. It is an application programming interface that provides database management system (DBMS) connectivity across a wide range of SQL databases, as well as access to other tabular data sources, such as spreadsheets or flat files.

Kilobytes/Second mode

The Kilobytes/Second mode is a dynamic load balancing mode that distributes connections based on which available server currently processes the fewest kilobytes per second.

last hop

A last hop is the final hop a connection takes to get to the BIG-IP system. You can allow the BIG-IP system to determine the last hop automatically to send packets back to the device from which they originated. You can also specify the last hop manually by making it a member of a last hop pool.

LDAP (Lightweight Directory Access Protocol)

LDAP is an Internet protocol that email programs use to look up contact information from a server.

Least Connections method

Least Connections method is a dynamic load balancing method that bases connection distribution on which server currently manages the fewest open connections.

link load balancing

Link load balancing is defined as managing traffic across multiple Internet or wide-area network (WAN) gateways.

load balancing method

A particular method of determining how to distribute connections across a load balancing pool.

load balancing pool

See *pool*.

load balancing virtual server

A load balancing virtual server is a virtual server that directs client traffic to a load balancing pool. This is the most basic type of virtual server. See also *virtual server*.

local traffic management

Local traffic management is the process of managing network traffic that comes into or goes out of a local area network (LAN), including an intranet.

loopback device

A loopback device is a software interface that is not associated with an actual network card. The nPath routing configuration requires you to configure loopback adapters on servers.

MAC (Media Access Control)

MAC is a protocol that defines the way workstations gain access to transmission media, and is most widely used in reference to LANs. For IEEE LANs, the MAC layer is the lower sublayer of the data link layer protocol.

MAC address

A MAC address is used to represent hardware devices on an Ethernet network.

member

Member is a reference to a node when it is included in a particular load balancing pool. Pools typically include multiple member nodes.

monitor

The Link Controller uses monitors to determine whether nodes are **up** or **down**. There are several different types of monitors and they use various methods to determine the status of a server or service.

monitor association

A monitor association is an association that a user makes between a health or performance monitor and a pool, pool member, or node.

monitor instance

You create a monitor instance when a health monitor is associated with a pool member or node. It is the monitor instance that actually performs the health check, not the monitor.

monitor template

A monitor template is an internal mechanism that the Link Controller uses to provide default values for a custom monitor when no pre-configured monitor exists.

MSRDP persistence

MSRDP persistence tracks sessions between clients and servers running the Microsoft® Remote Desktop Protocol (RDP) service.

multi-homed network

A multi-homed network is composed of one or more data centers that have more than one link to the Internet.

name resolution

Name resolution is the process by which a name server matches a domain name request to an IP address, and sends the information to the client requesting the resolution.

NAT (Network Address Translation)

A NAT is an alias IP address that identifies a specific node managed by the Link Controller to the external network.

network virtual server

A network virtual server is a virtual server whose IP address has no bits set in the host portion of the IP address (that is, the host portion of its IP address is **0**). There are two kinds of network virtual servers: those that direct client traffic based on a range of destination IP addresses, and those that direct client traffic based on specific destination IP addresses that the Link Controller does not recognize.

node

A node address is the IP address associated with one or more nodes. This IP address can be the real IP address of a network server, or it can be an alias IP address on a network server.

node status

Node status indicates whether a node is **up** and available to receive connections, or **down** and unavailable. The Link Controller uses the node ping and health check features to determine node status.

Observed method

Observed method is a dynamic load balancing method that bases connection distribution on a combination of two factors: the server that currently hosts the fewest connections and also has the fastest response time.

OCSP (Online Certificate Status Protocol)

OCSP is a protocol that authenticating systems can use to check on the revocation status of digitally-signed SSL certificates. The use of OCSP is an alternative to the use of a certificate revocation list (CRL). See also *certificate revocation list (CRL)*.

packet rate

The packet rate is the number of data packets per second processed by a server.

PAM (Pluggable Authentication Module)

A PAM module is a software module that a server application uses to authenticate client traffic. The modular design of a PAM module allows an organization to add, replace, or remove that authentication mechanism from a server application with minimal impact to that application. An example of a PAM module is an application that uses a remote Lightweight Directory Access Protocol (LDAP) server to authenticate client traffic. See also *LDAP (Lightweight Directory Access Protocol)*.

parent profile

A parent profile is a profile that can propagate its values to another profile. A parent profile can be either a default profile or a custom profile. See also *profile*.

passive failure

A passive failure consists of a server-connect failure or a failure to receive a data response within a user-specified interval.

performance monitor

A performance monitor gathers statistics and checks the state of a target device.

persistence

See *connection persistence* or *session persistence*.

persistence profile

A persistence profile is a configuration tool for implementing a specific type of session persistence. An example of a persistence profile type is a cookie persistence profile.

pipelining

Pipelining is a feature of HTTP/1.0 that allows clients to make requests even when prior requests have not yet received a response from the server.

pool

A pool is composed of a group of network devices (called members). The Link Controller load balances requests to the nodes within a pool based on the load balancing method and persistence method you choose when you create the pool or edit its properties.

pool member

A pool member is a server that is a member of a load balancing pool.

port

A port can be represented by a number that is associated with a specific service supported by a host. Refer to the Services and Port Index for a list of port numbers and corresponding services.

port-specific wildcard virtual server

A port-specific wildcard virtual server is a wildcard virtual server that uses a port number other than 0. See *wildcard virtual server*.

pre-configured monitor

A pre-configured monitor is a system-supplied health or performance monitor. You can use a pre-configured monitor as is, but you cannot modify or delete one. See also *monitor*.

Predictive method

Predictive method is a dynamic load balancing method that bases connection distribution on a combination of two factors: the server that currently hosts the fewest connections, and also has the fastest response time. Predictive method also ranks server performance over time, and passes connections to servers which exhibit an improvement in performance rather than a decline.

profile

A profile is a configuration tool containing settings for defining the behavior of network traffic. The Link Controller contains profiles for managing FastL4, HTTP, TCP, FTP, SSL, and RTSP traffic, as well as for implementing persistence and application authentication.

profile setting

A profile setting is a configuration attribute within a profile that has a value associated with it. You can configure a profile setting to customize the way that the Link Controller manages a type of traffic.

profile type

A profile type is a category of profile that you use for a specific purpose. An example of a profile type is an HTTP profile, which you configure to manage HTTP network traffic.

protocol profile

A protocol profile is a profile that you create for controlling the behavior of FastL4, TCP, UDP, and RTSP traffic.

Quality of Service (QoS) level

The Quality of Service (QoS) level is a means by which network equipment can identify and treat traffic differently based on an identifier. Essentially, the QoS level specified in a packet enforces a throughput policy for that packet.

RADIUS (Remote Authentication Dial-in User Service)

RADIUS is a service that performs remote user authentication and accounting. Its primary use is for Internet Service Providers, though it can also be used on any network that needs a centralized authentication and/or accounting service for its workstations.

rate class

You create a rate filter from the Configuration utility or command line utility. When you assign a rate class to a rate filter, a rate class determines the volume of traffic allowed through a rate filter. See also *rate shaping*.

rate shaping

Rate shaping is a type of extended IP filter. Rate shaping uses the same IP filter method but applies a rate class, which determines the volume of network traffic allowed. See also *rate class*.

ratio

A ratio is a parameter that assigns a weight to a virtual server for load balancing purposes.

Ratio method

The Ratio load balancing method distributes connections across an array of virtual servers in proportion to the ratio weights assigned to each individual virtual server.

Real-Time Stream Protocol (RTSP)

See *RTSP*.

receive expression

A receive expression is the text string that the Link Controller looks for in the web page returned by a web server during an extended content verification (ECV) health check.

redundant system configuration

Redundant system configuration refers to a pair of units that are configured for failover. In a redundant system configuration, there are two units, one running as the active unit and one running as the standby unit. If the active unit fails, the standby unit takes over and manages connection requests.

RFC 1918 addresses

An RFC 1918 address is an address that is within the range of non-routable addresses described in the IETF RFC 1918.

Round Robin mode

Round Robin mode is a static load balancing mode that bases connection distribution on a set server order. Round Robin mode sends a connection request to the next available server in the order.

RTSP

RTSP (Real-Time Streaming Protocol) establishes and controls one or more time-synchronized streams of continuous media such as audio or video.

Secure Network Address Translation (SNAT)

See *SNAT (Secure Network Address Translation)*. See also *intelligent SNAT*.

self IP address

Self IP addresses are the IP addresses owned by the BIG-IP system that you use to access the internal and external VLANs.

send string

A send string is the request that the Link Controller sends to the web server during an extended content verification (ECV) health check.

service

Service refers to services such as TCP, UDP, HTTP, and FTP.

services profile

A services profile is a configuration tool on the Link Controller for managing either HTTP or FTP network traffic.

session persistence

A series of related connections received from the same client, having the same session ID. When persistence is enabled, a Link Controller sends all connections having the same session ID to the same node, instead of load balancing the connections. Session persistence is not to be confused with *connection persistence*.

Setup utility

The Setup utility walks you through the initial system configuration process. You can run the Setup utility from the Configuration utility start page.

simple persistence

See *source address affinity persistence*.

SIP persistence

SIP persistence is a type of persistence used for servers that receive Session Initiation Protocol (SIP) messages sent through UDP. SIP is a protocol that enables real-time messaging, voice, data, and video.

SNAT (Secure Network Address Translation)

A SNAT is a feature you can configure on the Link Controller. A SNAT defines a routable alias IP address that one or more nodes can use as a source IP address when making connections to hosts on the external network. See also *standard SNAT*, *intelligent SNAT*.

SNAT pool

A SNAT pool is a pool of translation addresses that you can map to one or more original IP addresses. Translation addresses in a SNAT pool are not self-IP addresses.

SNMP (Simple Network Management Protocol)

SNMP is the Internet standard protocol, defined in STD 15, RFC 1157, developed to manage nodes on an IP network.

source address affinity persistence

Also known as simple persistence, source address affinity persistence supports TCP and UDP protocols, and directs session requests to the same server based solely on the source IP address of a packet.

SSL (Secure Sockets Layer)

SSL is a network communications protocol that uses public-key technology as a way to transmit data in a secure manner.

SSL persistence

SSL persistence is a type of persistence that tracks non-terminated SSL sessions, using the SSL session ID.

SSL profile

An SSL profile is a configuration tool that you use to terminate and initiate SSL connections from clients and servers.

standard SNAT

A standard SNAT is a SNAT that you implement by using the SNAT screens of the Configuration utility. See also *SNAT* and *intelligent SNAT*.

standby unit

A standby unit in a redundant system configuration is a unit that is always prepared to become the active unit if the active unit fails.

sticky persistence

See *destination address affinity persistence*.

TACACS (Terminal Access Controller Access Control System)

TACACS is an older authentication protocol common to UNIX systems. TACACS allows a remote access server to forward a user's login password to an authentication server.

TACACS+

TACACS+ is an authentication mechanism designed as a replacement for the older TACACS protocol. There is little similarity between the two protocols, however, and they are therefore not compatible.

Tcl

Tcl (Tools Command Language) is an industry-standard scripting language. On the Link Controller, users use Tcl to write iRules®.

transparent node

A transparent node appears as a router to other network devices, including the BIG-IP system.

Type of Service (ToS) level

The Type of Service (ToS) level is another means, in addition to the Quality of Service (QoS) level, by which network equipment can identify and treat traffic differently based on an identifier.

virtual address

A virtual address is an IP address associated with one or more virtual servers managed by the Link Controller.

virtual port

A virtual port is the port number or service name associated with one or more virtual servers managed by the Link Controller. A virtual port number should be the same TCP or UDP port number to which client programs expect to connect.

virtual server

Virtual servers are a specific combination of virtual address and virtual port, associated with a content site that is managed by a Link Controller or other type of host server.

VLAN

VLAN stands for virtual local area network. A VLAN is a logical grouping of network devices. You can use a VLAN to logically group devices that are on different network segments.

VLAN name

A VLAN name is the symbolic name used to identify a VLAN. For example, you might configure a VLAN named marketing, or a VLAN named development. See also *VLAN*.

wildcard virtual server

A wildcard virtual server is a virtual server that uses an IP address of **0.0.0.0**, * or "any". A wildcard virtual server accepts connection requests for destinations outside of the local network. Wildcard virtual servers are included only in Transparent Node Mode configurations.



Index

/etc/bigip.conf file 18-14

A

acceleration
 See hardware acceleration.
 address data groups 18-12
 agent types A-5
 algorithm
 See Nagle's algorithm.
 alias address 8-13
 alternate load balancing method, changing 9-13
 alternate methods, specifying 9-1
 application traffic, managing 13-1
 ARP requests 6-14

B

bandwidth
 borrowing 21-7, 21-8
 replenishing 21-5
 saving 21-5
 bandwidth size 3-6
 base packet rates 21-4, 21-7
 Base Rate setting, configuring 21-4
 base throughput rate 21-4
 BIG-IP link health monitor 8-9
 BIG-IP product line, introducing 1-1
 billing properties 3-6
 burst reservoirs 21-5
 Burst Size setting, configuring 21-4
 bursting restrictions 21-6

C

caching proxy servers 15-4
 Ceiling Rate setting, configuring 21-4
 ceiling rates 21-5, 21-8
 child rate classes 21-7
 chunking 14-4
 client IP addresses, tracking connections 15-5
 client traffic
 and rate classes 21-7
 directing subnet 6-3
 redirecting 2-5, 5-1
 clientside iRule context 18-7
 clone pool, and virtual servers 6-11
 commands
 and iRules 18-3
 and matchclass 18-11
 for data manipulation 18-9
 for data queries 18-9
 for function 18-9
 for statement 18-9
 for utility 18-9

Completion rate mode 9-6
 concurrent connection limits 5-18, 20-6
 concurrent connections 6-14
 connection limits
 allowing 6-14
 for nodes 5-18, 20-6
 connection pooling
 See X-Forwarded-For header.
 connection queueing 21-8
 Connection Rate mode 9-6
 connection requests, receiving 6-3
 connections, distributing by priority 5-15
 contains operator 18-11
 content switching, customizing 18-1
 context, iRules 18-7
 CPU metrics A-5
 custom HTTP profiles 13-5
 custom monitors
 importing from another custom monitor 8-5
 importing from pre-configured monitor 8-5
 importing from template 8-5
 custom profiles 13-4

D

data collection agents A-5
 data group members 18-16
 data group size 18-12
 data group storage
 See external data group storage.
 See in-line data group storage.
 data group types 18-12
 data groups
 configuring 18-11
 storing 18-15
 default HTTP profile 13-5, 14-2
 default profiles
 summarized 13-4
 using 13-2
 default wildcard virtual servers, creating 6-7
 destination address affinity persistence 15-4
 Destination Address Affinity profile settings 15-4
 destination address ranges
 and network virtual servers 6-3
 directing traffic to 6-3
 destination IP addresses, and persistence 15-4
 destination statement 10-3
 Direction setting, configuring 21-7
 disk metrics, gathering A-5
 DNS name resolution requests 9-1
 Drop Packet mode 9-3
 Duplex Billing option 3-6
 dynamic load balancing 9-1, 9-3, 9-6
 dynamic load balancing modes 9-6

- dynamic ratio
 - and Quality of Service mode 9-8
 - introducing 9-11
 - using with Quality of Service mode 9-11

- Dynamic Ratio mode
 - configuring RealSystem Servers for A-1
 - configuring VMI for A-3
 - described 5-13

E

- EAV monitors 8-2
- ECV monitors 8-2
- encoding, chunked and unchunked 14-4
- event declarations 18-6
- event execution, terminating 18-8
- event-based traffic management 18-6
- external data group storage 18-14
- external data groups 18-16

F

- failures of load balancing modes 9-1
- Fallback IP mode 9-3
- fallback load balancing method, changing 9-13
- Fast L4 profile settings 16-2
- Fastest mode, described 5-13
- FIFO
 - See PFIFO.
- firewalls 6-4
- forwarding virtual servers 6-2
- FQDNs, and wide IPs 7-1
- FTP health monitors 8-12
- FTP profile settings 14-6

G

- gateway ICMP health monitor 8-8
- Global Availability mode 9-4
- gtm_add script, running 11-5

H

- hardware acceleration 16-2
- hash persistence, defined 15-5
- health monitor settings 8-1
- health monitor types 8-1
- health monitors
 - and alias address 8-13
 - and association types 8-16
 - and BIG-IP link 8-9
 - and extended content verification 8-10
 - and link configuration 3-3
 - and pre-configured 8-4
 - and reverse mode 8-14
 - and transparent mode 8-14
 - and types of health monitors 8-2

- associating resource 8-15
- configuring 8-7, A-5
- creating 8-6
- creating custom 8-4
- defined 8-2
- deleting 8-17
- disabling 8-17
- displaying 8-17
- enabling 8-17
- for pools 5-8
- introducing 8-1
- logical grouping in 5-9, 5-19
- managing 8-17
- transparent mode in 5-9, 5-19
- using FTP 8-12
- using gateway ICMP 8-8
- using HTTP 8-11
- using HTTPS 8-11
- using ICMP 8-7
- using simple monitors 8-7
- using SNMP link 8-9
- using TCP 8-10
- using TCP Echo 8-8

- Hops mode 9-6
- host IP address data groups 18-12
- host virtual servers 6-3
- HTTP health monitor 8-11
- HTTP profile settings, configuring 14-2
- HTTP profiles
 - described 14-1
 - for default and custom 13-5
- HTTP traffic management 14-1
- HTTPS health monitor 8-11

I

- ICMP health monitor 8-7
- inbound load balancing
 - and load balancing modes 9-2
 - described 9-1
- in-line data group storage 18-14
- integer data groups 18-13
- intelligent SNATs 19-5
- internal interfaces 19-10
- internal network
 - See internal interfaces.
- IP address data groups 18-12
- IP address destinations 6-4
- IP addresses
 - and virtual servers 6-3
 - for clients 15-5
 - matching 6-3, 6-4
 - sharing 6-1
 - specifying for NATs 19-10
 - translating 6-4
- iRule command types 18-3

- iRule elements 18-2
- iRule evaluation, controlling 18-6
- iRule event declarations 18-2
- iRule event types 18-7
- iRule operators 18-2
- iRule prerequisites 18-6
- iRules
 - and virtual servers 18-5
 - assigning 6-16, 18-8
 - creating 18-5
 - defined 18-1
 - serverside context 18-7
- iRules statement commands 18-9

K

- kernel routing table 6-14
- Kilobyte/Second mode 9-7

L

- L2 forwarding virtual servers, defined 6-1
- last hop pools, and virtual servers 6-11
- LDNS round robin 9-13
- Least Connections load balancing method 5-14
- Least Connections mode 9-7
- linear white space, managing 14-5
- Link Capacity coefficient 9-14
- link configuration, and health monitors 3-3
- Link Controller modes 9-2
- link load balance reports, refining 3-6
- link load balancing 1-2
- link management 1-2
- link monitoring, defined 1-3
- link properties, configuring 2-3, 3-2
- link statistics 12-6
- link statistics reports, refining 3-6
- link weight properties 3-6
- Link Weighting screen 3-6
- links, selecting 9-1
- listeners
 - deleting 4-5
 - introducing 4-1
 - modifying 4-4
- load balancing
 - and dynamic modes 9-6
 - and static modes 9-3
 - using topology mode 10-5
- load balancing methods 5-1, 5-12
- load balancing mode failure 9-1
- load balancing mode types 9-2
- load balancing modes
 - for DNS name resolution 9-1
 - using Completion Rate 9-6
 - using connection rate 9-6
 - using Drop Packet 9-3
 - using Fallback IP 9-3

- using Global Availability 9-4
- using Hops 9-6
- using Kilobytes/Second 9-7
- using Least Connections 9-7
- using Packet Rate 9-7
- using Quality of Service 9-7, 9-8
- using Ratio 9-4
- using Round Robin 9-4
- using Round Trip Times (RTT) 9-7
- using Static Persist 9-4
- using Topology 9-5
- using Virtual Server Score 9-8
- using VS Capacity 9-8
- load balancing pools 2-5, 5-1
- load balancing virtual servers 6-1
- local DNS statistics 12-8
- logical operators 18-2
- longest match option 10-7

M

- masks, simple persistence 15-5
- matchclass command 18-11
- memory metrics, gathering A-5
- meta-data, external data groups 18-14
- min_active_members value 5-15
- minimum health monitors 20-5
- mode
 - load balancing 9-2
 - See also load balancing modes.
- monitor properties, and link configuration 3-3
- monitor types 8-2
- monitor-pool associations, managing 5-27
- monitors
 - for nodes 20-4
 - for pools 5-8
 - See also health monitors.
- multi-homed networks 1-2

N

- Nagle's algorithm 16-6
- NATs
 - configuring 19-10
 - described 19-10
- netmasks, specifying 6-7
- network IP address data groups 18-12
- network traffic, managing 13-1
- network virtual server types 6-3
- network virtual servers 6-3
- node configuration 19-10
- nodes
 - and connection limits 5-18, 20-6
 - and status icons 20-8
 - as pool members 5-7
 - defined 20-1
 - receiving connections 5-12

NTP, synchronization 11-2
 numeric value classes 18-13

O

Observed mode, described 5-14
 operators 18-2
 order, packets 21-1
 outbound traffic, and ToS level 5-11

P

packet filters 21-9
 packet order 21-1, 21-8
 packet rate limit, specifying 21-4
 Packet Rate mode 9-7
 packet rate, exceeding 21-4
 packet scheduling methods 21-2
 packet throughput, enforcing 21-1
 Packet Velocity ASIC 16-2
 packets, queuing and dequeuing 21-1, 21-8
 Parent Class setting, configuring 21-7
 parent HTTP profiles, specifying 14-4
 parent profiles, defined 13-4
 parent rate classes, borrowing bandwidth from 21-7
 path statistics 12-7
 performance (Layer 4) virtual servers 6-2
 persist command for iRules 18-11
 Persist mode, static 9-4
 persistence
 and conditions 15-3
 and iRules 18-1, 18-11
 and need for 15-1
 See also destination address affinity persistence.
 See also session persistence.
 persistence profile types 15-2
 persistence profiles 13-1
 persistence timer 15-5
 persistence, hash defined 15-5
 PFIFO 21-8
 pool member status 5-24
 pool members
 adding 5-16
 as servers 5-7
 checking status 5-24
 defined 5-1
 selecting with iRules 18-4
 pool monitoring 5-8
 pool naming 5-7, 19-7
 pool settings, and default values 5-16
 pool status 5-21
 pool-monitor associations, managing 5-27
 pools
 and SNAT/NAT connections 5-9
 and wide IPs 7-1
 checking status 5-21
 defined 2-5, 5-1

 deleting 5-22, 5-27
 selecting with iRules 18-1, 18-3
 port translation, turning off 6-7, 6-8
 port-specific wildcard virtual servers, creating 6-7, 6-8
 Predictive mode, described 5-14
 preferred load balancing method, changing 9-13
 preferred methods, described 9-1
 price weighting 3-6
 Priority FIFO
 See PFIFO.
 priority member activation 5-15
 priority numbers, assigning 5-15
 profile dependencies 13-11
 profile names, specifying 14-3
 profile settings, overriding 18-10
 profile summary 13-4
 profile types 13-1
 profiles
 and types 13-1
 and virtual servers 6-10, 13-10
 introducing 13-1
 using 13-2
 protocol profiles 13-1
 protocols
 and persistence settings 15-5
 and virtual servers 6-10
 proxy servers 6-4
 PVA hardware acceleration 16-2

Q

QOS coefficients
 adjusting 9-14
 overview 9-9
 QoS level, setting 5-11
 QoS pool attribute 5-11
 Quality of Service mode
 and default settings 9-8
 customizing 9-8, 9-10
 definition 9-7
 introducing 9-8
 understanding QOS coefficients 9-9
 using dynamic ratio 9-8, 9-11
 Queue Discipline setting, configuring 21-8

R

rate class example 21-6
 Rate Class settings 21-3
 rate classes
 and direction 21-7
 assigning 21-2
 creating 21-2
 defined 21-1, 21-2
 managing 21-9
 naming 21-4
 rate shaping, defined 21-1

rate, packets 21-1
 Ratio method, described 5-13
 Ratio mode 9-4
 ratio weighting 3-6
 ratio weights, specifying 5-17
 RealSystem Servers, configuring for load balancing A-1
 records, topology 10-3
 regions 10-6
 relational operators, listed 18-2
 request source statements 10-3
 requests
 chunking and unchunking 14-4
 distributing 9-3
 reservoirs
 See burst reservoirs.
 resource availability 8-2
 responses, chunking and unchunking 14-4
 reverse mode 8-14
 Round Robin mode
 and default load balancing 5-12
 definition 9-4
 Round Trip Times mode 9-7
 routable IP addresses 19-2
 route advertisement 6-13
 route insertion 6-14
 routers 6-4
 routing table 6-14
 rule operators 18-2
 rules
 See iRules.

S

Search box 13-9
 self-IP addresses, assigning 6-7
 server availability, increasing 6-1
 server overload 5-1
 server traffic, and rate classes 21-7
 servers
 and NTP 11-2
 selecting with iRules 18-4
 serverside iRule context 18-7
 services profiles 13-1
 session persistence
 and iRules 18-11
 enabling 6-1
 settings, for Protocol profiles 16-1
 SFQ 21-8
 simple monitors 8-2
 simple persistence
 See source address affinity persistence.
 SNAT pools
 and virtual servers 6-11
 assigning to virtual server 19-9
 snatpool command 18-4, 19-5
 SNATs, enabling and disabling 5-9

SNMP DCA Base monitor A-5
 SNMP DCA monitor A-5
 SNMP link health monitor 8-9
 source address affinity persistence 15-5
 Source Address Affinity persistence profile settings 15-6
 source IP addresses 19-10
 standard SNATs 19-5
 statement commands
 defined 18-3
 specifying 18-9
 static load balancing 9-1
 static load balancing modes 9-3
 Static Persist mode 9-4
 statistics
 accessing 12-1
 and links 12-6
 and local DNS servers 12-8
 and paths 12-7
 and types 12-3
 and wide IPs 12-3
 for virtual servers 6-17
 introducing 12-1
 Statistics profile 17-2
 status icons
 for nodes 20-8
 for pool members 5-24
 for pools 5-22
 for virtual servers and addresses 6-19
 sticky persistence
 See destination address affinity persistence.
 sticky persistence type 15-4
 Stochastic Fair Queueing
 See SFQ.
 string data groups 18-13
 sync groups
 See synchronization.
 synchronization
 activating 11-3
 and NTP 11-2
 and time 11-2
 controlling 11-3
 creating groups 11-4
 deactivating 11-4
 defined 11-1
 synchronization groups 11-4
 system resources
 associating health monitors to 8-15

T

Tcl syntax 18-1
 TCP Echo health monitor 8-8
 TCP health monitor 8-10
 throughput limitations 21-1, 21-2
 throughput policies, enforcing 21-1
 throughput rates 21-4

- throughput restrictions, applying 21-7
- throughput, enforcing 21-1
- timeout values 15-5
- Tools Command Language syntax
 - See Tcl syntax.
- topologies
 - and destination statements 10-3
 - and longest match option 10-7
 - and records 10-3
 - and regions 10-6
 - and request source statements 10-3
 - and wide IPs 10-5
 - introducing 10-1
 - setting up 10-3
- Topology mode 9-5
- topology records
 - introducing 10-3
 - removing 10-4
- ToS field, queueing 21-8
- ToS level, setting 5-11
- ToS pool attribute 5-11
- traffic
 - and QoS level 5-11
 - distributing by priority 5-15
 - managing 13-1
 - oversaturating 3-6
 - queueing 21-8
- traffic acceleration 16-2
- traffic across links, managing 3-6
- traffic direction, and rate classes 21-1
- traffic flow limits 21-4
- traffic flow rates 21-4, 21-5, 21-7
- traffic queueing 21-8
- traffic rates, bursting 21-4
- traffic types, managing 6-1
- translation address properties 19-7, 19-8
- translation addresses, choosing 19-5
- transparent device pools, creating 6-7
- transparent devices, receiving connections from 6-5
- transparent mode 5-9, 5-19, 8-14
- transparent nodes 6-4, 6-5
- Type of Service field
 - See ToS field.

U

- UC Davis agent A-5
- UDP profile settings 16-9
- UIE function commands, listed 18-9
- unrecognized destination addresses 6-3
- unused bandwidth
 - borrowing 21-7
 - replenishing 21-5
 - saving 21-4
- Use Price Weighting option 3-6
- Use Ratio Weighting option 3-6

- user-defined metrics, gathering A-5
- utility commands, defined 18-3

V

- virtual server addresses, and VLANs 6-7
- virtual server capabilities 6-1
- virtual server lists
 - returning 9-13
- virtual server mappings, defining wildcard 6-8
- virtual server properties, configuring 6-9
- virtual server resources
 - assigning 6-9
 - modifying 6-16
- Virtual Server Score mode 9-8
- virtual server settings 6-9
- virtual server statistics, viewing 6-17
- virtual server types 6-3
- virtual servers
 - and iRules 6-1, 18-8
 - and persistence 15-3
 - and profiles 6-10, 13-10
 - defined 6-1
 - deleting 6-20
 - disabling 6-7
 - for Fast L4 profile 6-2
 - forwarding 19-11
 - grouping 7-1
 - See also L2 forwarding virtual servers.
 - selecting 9-1
 - viewing 6-17
- VLAN groups, creating 6-7
- VS Capacity mode 9-8

W

- when keyword 18-8
- wide IP pools 9-1
- wide IP settings, modifying 7-3
- wide IP statistics 12-3
- wide IPs
 - adding 2-7
 - and topology load balancing 10-5
 - configuring 7-1
 - introducing 7-1
 - maintaining 7-2
- wildcard characters
 - and wide IPs 7-2
 - examples of 7-2
- wildcard servers
 - assigning to VLANs 6-5
 - creating 6-7
- wildcard virtual servers 6-4
- Windows 2000 Server agent A-5
- WMI, configuring for dynamic ratio load balancing A-3

X

X-Forwarded-For header 14-5