Acknowledgments
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 Chistos Zoulas.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.
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 Niels Mueller <nisse@lysator.liu.se>, which is protected under the GNU Public License.

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 (eay@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.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# Table of Contents

---

- [ ]

---

---
# Configuration Guide for BIG-IP® Local Traffic Management

## 1 Introducing Local Traffic Management

Understanding local traffic management ................................................................. 1-1
Summary of local traffic-management capabilities ............................................. 1-1
Managing specific types of application traffic .................................................... 1-2
Optimizing performance ................................................................. 1-3
Enhancing network security ......................................................... 1-5
Summarizing local traffic management configuration ...................................... 1-7
Using the Configuration utility ........................................................................ 1-7
Configuring local traffic objects ...................................................................... 1-7
Managing timeout settings for connections and sessions ............................... 1-9
Getting started ....................................................................................... 1-11
Using the network map feature ................................................................ .. 1-14
Understanding the filtering mechanism ....................................................... 1-14
Displaying a summary of local traffic objects .............................................. 1-15
Displaying the network map ........................................................................ 1-17
Understanding application templates .............................................................. 1-19
Benefits of application templates ................................................................. 1-19
Working with application templates ............................................................... 1-19
About this guide .................................................................................... 1-23
Finding additional information .................................................................... 1-23
Stylistic conventions .................................................................................. 1-24
Finding help and technical support resources ............................................... 1-26

## 2 Configuring Virtual Servers

Introducing virtual servers and virtual addresses ................................................ 2-1
What is a virtual server? .................................................................................. 2-1
What is a virtual address? .............................................................................. 2-2
How does a virtual server work? ................................................................. 2-3
Understanding virtual server types ............................................................... 2-5
Host virtual servers .................................................................................. 2-5
Network virtual servers ............................................................................. 2-5
Understanding clustered multi-processing .................................................... 2-8
Seeing differences in system behavior when CMP is enabled ....................... 2-8
Disabling and re-enabling CMP ................................................................. 2-9
Understanding CMP on redundant systems ............................................. 2-10
Creating a virtual server ............................................................................... 2-11
Understanding virtual server and virtual address settings .......................... 2-14
Configuring virtual server settings ............................................................. 2-14
Configuring virtual address settings ......................................................... 2-20
Managing virtual servers and virtual addresses ............................................ 2-22
Viewing or modifying a virtual server configuration .................................... 2-22
Viewing or modifying a virtual address configuration .................................. 2-24
Understanding virtual server and virtual address status ............................... 2-25
Enabling or disabling a virtual server or virtual address ............................ 2-26
Deleting a virtual server or virtual address .................................................. 2-27
# Table of Contents

## 3 Configuring Nodes

- Introducing nodes ........................................................................................................... 3-1
- Creating and modifying nodes ......................................................................................... 3-2
- Configuring node settings ............................................................................................... 3-4
  - Specifying an address for a node .................................................................................. 3-5
  - Specifying a node name ................................................................................................ 3-5
  - Assigning health monitors ............................................................................................ 3-5
  - Specifying the availability requirement ........................................................................ 3-7
  - Specifying a ratio weight ............................................................................................... 3-7
  - Setting a connection limit ............................................................................................. 3-7
- Managing nodes ................................................................................................................ 3-8
  - Viewing a list of nodes .................................................................................................. 3-8
  - Viewing node properties ............................................................................................... 3-8
  - Understanding node status ......................................................................................... 3-9
  - Enabling or disabling a node ....................................................................................... 3-11
  - Deleting a node ........................................................................................................... 3-11
  - Removing monitor associations ................................................................................... 3-11

## 4 Configuring Load Balancing Pools

- Introducing load balancing pools .................................................................................... 4-1
- What is a load balancing pool? ........................................................................................ 4-1
- Features of a load balancing pool .................................................................................... 4-1
- Creating and modifying load balancing pools ................................................................ 4-2
  - Creating and implementing a load balancing pool ....................................................... 4-2
  - Modifying a load balancing pool .................................................................................. 4-3
  - Modifying pool membership ....................................................................................... 4-4
- Configuring pool settings ............................................................................................... 4-6
  - Specifying a pool name ............................................................................................... 4-7
  - Associating health monitors with a pool ..................................................................... 4-7
  - Specifying the availability requirements ..................................................................... 4-8
  - Allowing SNATs and NATs ............................................................................................ 4-9
  - Specifying action when a service becomes unavailable .............................................. 4-9
  - Configuring a slow ramp time ..................................................................................... 4-9
  - Configuring the Type of Service (ToS) level ............................................................... 4-10
  - Configuring the Quality of Service (QoS) level ......................................................... 4-10
  - Specifying the load balancing method ........................................................................ 4-11
  - Specifying priority-based member activation ............................................................. 4-14
  - Specifying pool members ............................................................................................ 4-15
- Configuring pool member settings .................................................................................. 4-16
  - Specifying a ratio weight for a pool member ............................................................... 4-17
  - Specifying priority-based member activation ............................................................. 4-17
  - Specifying a connection limit ..................................................................................... 4-17
  - Selecting an explicit monitor association ..................................................................... 4-17
  - Creating an explicit monitor association for a pool member ....................................... 4-18
  - Specifying an availability requirement ....................................................................... 4-19
- Managing pools and pool members .............................................................................. 4-20
  - Managing pools ........................................................................................................... 4-20
  - Managing pool members ............................................................................................. 4-23
  - Removing monitor associations ................................................................................... 4-28
  - Viewing pool and pool member statistics ................................................................... 4-28
5
Understanding Profiles

Introducing profiles ................................................................. 5-1
Profile types ................................................................. 5-1
Default profiles ............................................................ 5-3
Custom and parent profiles ........................................... 5-4
Summarizing profiles ...................................................... 5-5
Creating and modifying profiles ........................................ 5-6
Using a default profile as is ............................................ 5-6
Modifying a default profile ............................................ 5-7
Creating a custom profile ............................................. 5-8
Modifying a custom profile ............................................ 5-9
Viewing and deleting profiles ........................................ 5-11
Viewing a list of profiles ................................................ 5-11
Deleting a profile .......................................................... 5-12
Implementing a profile .................................................... 5-12
Finding more information ............................................. 5-13

6
Managing Application Layer Traffic

Introducing application layer traffic management ..................... 6-1
Configuring HTTP standard profile settings ............................... 6-2
Understanding HTTP profile settings ........................................ 6-2
Configuring HTTP compression ............................................... 6-12
Compression in a typical client-server scenario ......................... 6-12
Compression using the BIG-IP system ......................................... 6-12
Using an HTTP profile for data compression ............................. 6-14
Working with data compression strategies ................................. 6-23
Understanding compression providers .................................... 6-23
Understanding compression strategy selection ....................... 6-24
Understanding the adaptive compression strategy ..................... 6-24
Viewing compression statistics ........................................... 6-28
Configuring the RAM Cache feature ......................................... 6-30
Getting started with RAM caching ........................................ 6-30
Understanding RAM Cache settings ....................................... 6-34
Using an HTTP profile for RAM caching ................................. 6-34
Optimizing data compression and RAM caching ......................... 6-36
Using the http-acceleration profile ........................................ 6-36
Using the http-wan-optimized-compression profile .................. 6-37
Using the http-lan-optimized-caching profile .......................... 6-38
Using the http-wan-optimized-compression-caching profile ........ 6-39
Configuring FTP profile settings ........................................ 6-40
Specifying a profile name ................................................ 6-40
Specifying a parent profile ............................................. 6-41
Specifying a Translate Extended value ................................... 6-41
Specifying a data port .................................................. 6-41
Enabling security for FTP traffic ......................................... 6-41
Configuring SIP profile settings ........................................ 6-42
Understanding SIP profile settings ....................................... 6-42
Customizing SIP persistence .................................................. 6-46
Configuring RTSP profile settings ........................................ 6-47
Configuring iSession profile settings ....................................... 6-50
Table of Contents

7
Enabling Session Persistence

- Introducing session persistence ................................................................. 7-1
- Configuring a persistence profile ............................................................... 7-1
- Enabling session persistence through iRules ........................................... 7-2
- Using the OneConnect profile with session persistence ......................... 7-2
- Persistence types and their profiles ...................................................... 7-4
  - Types of persistence ........................................................................... 7-4
  - Understanding criteria for session persistence ................................ 7-5
  - Cookie persistence ........................................................................... 7-7
  - Destination address affinity persistence ........................................... 7-11
  - Hash persistence ............................................................................ 7-12
  - Microsoft Remote Desktop Protocol persistence ............................ 7-14
  - SIP persistence ............................................................................. 7-18
  - Source address affinity persistence ................................................ 7-20
  - SSL persistence .......................................................................... 7-21
  - Universal persistence ...................................................................... 7-22

8
Managing Protocol Profiles

- Introducing protocol profiles ................................................................. 8-1
- Configuring a Fast L4 profile ................................................................. 8-2
  - Understanding Fast L4 profile settings ........................................ 8-2
  - Configuring PVA hardware acceleration .......................................... 8-5
- Configuring a Fast HTTP profile .......................................................... 8-7
  - Understanding Fast HTTP profile settings .................................... 8-8
- Configuring an HTTP Class profile ...................................................... 8-11
  - Understanding HTTP Class profile settings ................................. 8-11
  - Classifying traffic through other BIG-IP modules ........................ 8-14
- Configuring TCP profiles ................................................................. 8-16
  - Understanding TCP profile settings ........................................... 8-16
  - Understanding tcp-lan-optimized profile settings ....................... 8-21
  - Understanding tcp-wan-optimized profile settings ..................... 8-22
- Configuring a UDP profile ................................................................. 8-24
- Configuring an SCTP profile ............................................................... 8-25

9
Managing SSL Traffic

- Introducing SSL traffic management .................................................. 9-1
  - Managing client-side and server-side traffic .................................. 9-1
  - Summarizing SSL traffic-control features ...................................... 9-2
  - Understanding certificate verification ............................................. 9-3
  - Understanding certificate revocation .............................................. 9-5
  - Understanding encryption/decryption .......................................... 9-5
  - Understanding client authorization .............................................. 9-6
  - Understanding SSL session persistence ....................................... 9-6
  - Understanding other SSL features .............................................. 9-6
- Managing keys and certificates ...................................................... 9-7
  - Displaying information about existing keys and certificates .......... 9-7
  - Creating a request for a new certificate and key ............................ 9-8
  - Renewing a certificate ................................................................. 9-9
  - Deleting a certificate/key pair ...................................................... 9-10
  - Importing keys, certificates, and archives .................................... 9-10
  - Creating an archive ..................................................................... 9-11
## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding SSL profiles</td>
<td>9-11</td>
</tr>
<tr>
<td>Configuring SSL profile settings</td>
<td>9-13</td>
</tr>
<tr>
<td>Specifying a profile name</td>
<td>9-14</td>
</tr>
<tr>
<td>Selecting a parent profile</td>
<td>9-14</td>
</tr>
<tr>
<td>Specifying a certificate name</td>
<td>9-15</td>
</tr>
<tr>
<td>Specifying a key name</td>
<td>9-15</td>
</tr>
<tr>
<td>Specifying and confirming a pass phrase</td>
<td>9-15</td>
</tr>
<tr>
<td>Configuring a certificate chain</td>
<td>9-15</td>
</tr>
<tr>
<td>Specifying trusted client CAs</td>
<td>9-16</td>
</tr>
<tr>
<td>Specifying SSL ciphers</td>
<td>9-16</td>
</tr>
<tr>
<td>Configuring workarounds</td>
<td>9-17</td>
</tr>
<tr>
<td>Enabling ModSSL method emulation</td>
<td>9-20</td>
</tr>
<tr>
<td>Configuring the SSL session cache</td>
<td>9-21</td>
</tr>
<tr>
<td>Specifying an alert timeout value</td>
<td>9-22</td>
</tr>
<tr>
<td>Specifying a handshake timeout value</td>
<td>9-22</td>
</tr>
<tr>
<td>Forcing renegotiation of SSL sessions</td>
<td>9-22</td>
</tr>
<tr>
<td>Configuring SSL shutdowns</td>
<td>9-23</td>
</tr>
<tr>
<td>Accepting non-SSL connections</td>
<td>9-24</td>
</tr>
<tr>
<td>Configuring client and server authentication settings</td>
<td>9-25</td>
</tr>
<tr>
<td>Configuring certificate presentation</td>
<td>9-26</td>
</tr>
<tr>
<td>Configuring per-session authentication</td>
<td>9-27</td>
</tr>
<tr>
<td>Advertising a list of trusted client CAs</td>
<td>9-28</td>
</tr>
<tr>
<td>Configuring authentication depth</td>
<td>9-28</td>
</tr>
<tr>
<td>Configuring name-based authentication</td>
<td>9-29</td>
</tr>
<tr>
<td>Certificate revocation</td>
<td>9-29</td>
</tr>
</tbody>
</table>

### 10 Authenticating Application Traffic

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing remote authentication</td>
<td>10-1</td>
</tr>
<tr>
<td>BIG-IP system authentication modules</td>
<td>10-1</td>
</tr>
<tr>
<td>Implementing authentication modules</td>
<td>10-2</td>
</tr>
<tr>
<td>Implementing an LDAP authentication module</td>
<td>10-4</td>
</tr>
<tr>
<td>Creating an LDAP configuration object</td>
<td>10-5</td>
</tr>
<tr>
<td>Creating an LDAP profile</td>
<td>10-7</td>
</tr>
<tr>
<td>Implementing a RADIUS authentication module</td>
<td>10-10</td>
</tr>
<tr>
<td>Creating a RADIUS server object</td>
<td>10-10</td>
</tr>
<tr>
<td>Creating a RADIUS configuration object</td>
<td>10-11</td>
</tr>
<tr>
<td>Creating a RADIUS profile</td>
<td>10-13</td>
</tr>
<tr>
<td>Implementing a TACACS+ authentication module</td>
<td>10-16</td>
</tr>
<tr>
<td>Creating a TACACS+ configuration object</td>
<td>10-16</td>
</tr>
<tr>
<td>Creating a TACACS+ profile</td>
<td>10-18</td>
</tr>
<tr>
<td>Implementing an SSL client certificate LDAP authentication module</td>
<td>10-21</td>
</tr>
<tr>
<td>Understanding SSL client certificate authorization</td>
<td>10-21</td>
</tr>
<tr>
<td>Creating an SSL client certificate LDAP configuration object</td>
<td>10-24</td>
</tr>
<tr>
<td>Creating an SSL client certificate LDAP authorization profile</td>
<td>10-27</td>
</tr>
<tr>
<td>Implementing an SSL OCSP authentication module</td>
<td>10-30</td>
</tr>
<tr>
<td>Understanding OCSP</td>
<td>10-30</td>
</tr>
<tr>
<td>Creating an OCSP responder object</td>
<td>10-33</td>
</tr>
<tr>
<td>Creating an SSL OCSP configuration object</td>
<td>10-36</td>
</tr>
<tr>
<td>Creating an SSL OCSP profile</td>
<td>10-37</td>
</tr>
<tr>
<td>Implementing a CRLDP authentication module</td>
<td>10-40</td>
</tr>
<tr>
<td>Creating a CRLDP server object</td>
<td>10-41</td>
</tr>
<tr>
<td>Creating a CRLDP configuration object</td>
<td>10-42</td>
</tr>
<tr>
<td>Creating a CRLDP profile</td>
<td>10-43</td>
</tr>
</tbody>
</table>
15 Configuring Traffic Classes

Introducing traffic classes ................................................................. 15-1
Creating a traffic class ................................................................. 15-1
Configuring traffic class settings .................................................. 15-2
Managing traffic classes ............................................................... 15-3

16 Configuring Rate Shaping

Introducing rate shaping ................................................................. 16-1
Creating and implementing rate classes ....................................... 16-2
Configuring rate class settings ..................................................... 16-4
Specifying a name ........................................................................ 16-5
Specifying a base rate ................................................................. 16-5
Specifying a ceiling ................................................................. 16-5
Specifying a burst size .............................................................. 16-6
Specifying direction ................................................................. 16-8
Specifying a parent class ......................................................... 16-8
Specifying a shaping policy ...................................................... 16-9
Specifying a queue method ...................................................... 16-9
Specifying a drop policy ........................................................... 16-10
Managing rate classes ............................................................... 16-11

17 Writing iRules

Introducing iRules ........................................................................... 17-1
What is an iRule? .......................................................................... 17-1
Basic iRule elements ................................................................... 17-2
Specifying traffic destinations and address translations ............ 17-4
Creating iRules ........................................................................... 17-7
Understanding iRules and administrative partitions .................. 17-7
Controlling iRule evaluation ....................................................... 17-8
Configuration prerequisites ....................................................... 17-8
Specifying events ........................................................................ 17-8
Using iRule commands ............................................................. 17-11
Statement commands ................................................................. 17-11
Query and manipulation commands ......................................... 17-11
Utility commands ....................................................................... 17-11
Working with profiles ............................................................... 17-12
Reading profile settings ............................................................ 17-12
Overriding profile settings ........................................................ 17-12
# Table of Contents

Enabling session persistence with iRules ....................................................... 17-13
Creating, managing, and using data groups ................................................ 17-14
  Using the matchclass command ................................................................. 17-14
Creating data groups .................................................................................. 17-15
Storage options ........................................................................................... 17-17
Displaying data group properties ................................................................. 17-18
Managing data group members ................................................................. 17-19

A

Health and Performance Monitors
  Monitor reference ........................................................................................ A-1
  Virtual server considerations ................................................................. A-20
  CMP considerations ................................................................................. A-21
  Prerequisite tasks for MSSQL ................................................................. A-27
  MSSQL monitor settings and their default values .................................. A-27
  Troubleshooting MSSQL logins .............................................................. A-29

B

Additional Monitor Considerations
  Implementing monitors for Dynamic Ratio load balancing ...................... B-1
  Implementing a Real Server monitor ....................................................... B-1
  Implementing a WMI monitor ................................................................. B-3
  Implementing an SNMP DCA or SNMP DCA Base monitor .................. B-10
  Implementing an MSSQL monitor .......................................................... B-11

C

Working with the OpenSSL Utility
  Introducing OpenSSL for the BIG-IP system .......................................... C-1
  Creating a certificate for a client system .................................................. C-1
  Creating a certificate for a web site ......................................................... C-2
  Working with certificate revocation ....................................................... C-3
  Performing other certificate-related tasks ............................................. C-4

Glossary

Index
Introducing Local Traffic Management

- Understanding local traffic management
- Summarizing local traffic management configuration
- Using the network map feature
- Understanding application templates
- About this guide
- Finding help and technical support resources
Understanding local traffic management

The BIG-IP® local traffic management system is specifically designed to manage your local network traffic. **Local traffic management** refers to the process of managing network traffic that comes into or goes out of a local area network (LAN), including an intranet.

This configuration guide applies to the set of local traffic management products that are part of the BIG-IP system family of products.

A commonly-used feature of the BIG-IP system is its ability to intercept and redirect incoming network traffic, for the purpose of intelligently tuning the load on network servers. However, tuning server load is not the only type of local traffic management. The BIG-IP system includes a variety of features that perform functions such as inspecting and transforming header and content data, managing SSL certificate-based authentication, and compressing HTTP responses. In so doing, the BIG-IP system not only directs traffic to the appropriate server resource, but also enhances network security and frees up server resources by performing tasks that web servers typically perform.

**Note**

**BIG-IP® Local Traffic Manager is one of several products that constitute the BIG-IP product family. All products in the BIG-IP product family run on the powerful Traffic Management Operating System, commonly referred to as TMOSTM. For an overview of the complete BIG-IP product offering, see the introductory chapter of the** **TMOSTM Management Guide for BIG-IP® Systems. For information on getting started with BIG-IP system configuration, see the** **BIG-IP® Systems: Getting Started Guide.**

Summary of local traffic-management capabilities

When configured properly, the BIG-IP system can perform a wide variety of traffic-management functions, such as:

- Balancing traffic to tune and distribute server load on the network for scalability.
- Off-loading standard server tasks, such as HTTP data compression, SSL authentication, and SSL encryption to improve server performance.
- Monitoring the health and performance of servers on the network for availability.
- Establishing and managing session and connection persistence.
- Handling application-traffic authentication and authorization functions based on user name/password and SSL certificate credentials.
- Managing packet throughput to optimize performance for specific types of connections.
• Improving performance by aggregating multiple client requests into a server-side connection pool. This aggregation of client requests is part of the BIG-IP system’s OneConnect™ feature.
• Applying configuration settings to customize the flow of application-specific traffic (such as HTTP and SSL traffic).
• Customizing the management of specific connections according to user-written scripts based on the industry-standard Tool Command Language (Tcl).
• For multi-processor platforms, enhancing traffic-management performance by configuring the system to process traffic using multiple instances of the system's Traffic Management Microkernel (TMM) service.

While some of the functions on this list offer the basic ability to balance the load on your network servers, other functions on the list offer specialized abilities that are worth noting. These abilities include managing specific types of application traffic, optimizing server performance, and enhancing the security of your network. The following three sections describe these specialized capabilities.

Managing specific types of application traffic

Applying configuration settings to customize the flow of application-specific traffic is a key feature of local traffic management. The BIG-IP system can control many different kinds of traffic, each in a different way. You do this by establishing a policy for managing each type of network traffic. Examples of traffic types that the system can manage are: TCP, UDP, HTTP, FTP, SSL, and Session Initiation Protocol (SIP).

In addition to creating separate policies to systematically manage these different traffic types, you can also do the following tasks:

• Write iRules™ to assign certain behaviors to individual application-specific connections.
  iRules can search the content of a particular type of traffic, such as an HTTP request or response, and direct the traffic accordingly.
• Insert header data into application-specific requests.
  You can insert header data into traffic such as HTTP requests, and then direct the request based on that header data.
• Implement session persistence.
  Using the BIG-IP system’s powerful configuration tools, you can configure session persistence, based on data such as HTTP cookies, source IP addresses, destination IP addresses, and SSL session IDs.
• Monitor the health or performance of nodes or pool members.
  For example, the BIG-IP system can monitor web servers on a network, and if the system determines that a target web server is non-functional, the BIG-IP system can redirect the request to a different server.
• **Use the dynamic ratio load-balancing algorithm.**

This algorithm assesses the current load on a particular type of server, such as a Windows Management Infrastructure (WMI) server, and then redirect a request based on that assessment. The ability to monitor servers corresponding to specific types of applications is a key tool for maintaining optimal performance of your network.

Finally, the BIG-IP system offers a set of application templates for you to use. Each template provides wizard-like screens for configuring the system to process traffic for a specific application. Examples of templates that the BIG-IP system provides are BEA WebLogic and Microsoft® IIS.

### Optimizing performance

The BIG-IP system includes several features designed to optimize server performance. Such features either offload labor-intensive traffic management tasks, such as SSL certificate verification, or enable the pooling, re-use, and overall persistence of server-side connections.

### Offloading server tasks

The BIG-IP system can offload these tasks from a network server:

- SSL certificate-based authentication, including the checking of certificate revocation status through the use of certificate revocation lists (CRLs), Online Certificate Status Protocol (OCSP), or Certificate Revocation List Distribution Point (CRLDP) technology.
- SSL encryption and decryption
- SSL certificate-based authorization using remote LDAP servers
- HTTP data compression and RAM caching
- Authentication and authorization of application traffic when using remote authentication servers such as LDAP and RADIUS servers
- The rewriting of Microsoft® Remote Desktop connections

### Optimizing TCP and HTTP connections

The BIG-IP system manages TCP and HTTP connections in certain ways to optimize server performance. Primary network optimization features are: OneConnect™, HTTP pipelining, HTTP data compression, RAM caching, and rate shaping.

### OneConnect

The OneConnect™ feature contains the following components:

- **Content Switching**

  When an HTTP client sends multiple requests within a single connection, the BIG-IP system is able to process each of those requests individually, sending those requests to different destination servers if necessary.
- **Connection Pooling**
  With this feature, the BIG-IP system combines server-side connections that remain open but are not in use, so that other clients can use them. This can significantly reduce the number of servers required to process client requests.

- **OneConnect transformation**
  Sometimes, for HTTP/1.0 requests, you might want to add Keep-Alive support to HTTP `Connection` headers, to ensure that server-side connections remain open. This manipulation of HTTP `Connection` headers is a feature known as OneConnect transformation. This feature works in conjunction with connection pooling.

  For more information on the OneConnect feature in general, see Chapter 6, *Managing Application Layer Traffic*, and Chapter 11, *Using Additional Profiles*.

**HTTP pipelining**

In addition to the OneConnect feature, the BIG-IP system has the ability to process pipelined requests. This means that the BIG-IP system can initiate multiple client requests without waiting for the previous request to receive a response. Pipelining is an optimization feature available for HTTP/1.1 requests only.

For more information on HTTP pipelining, see Chapter 6, *Managing Application Layer Traffic*.

**HTTP data compression**

To reduce the load on your back-end servers, you can configure an HTTP profile to compress HTTP responses. When the BIG-IP system compresses HTTP responses, the back-end servers processing the HTTP traffic no longer need to use resources to perform data compression.

If you have a Local Traffic Manager system on either side of a wide-area network (WAN), you can configure an iSession profile. This creates an optimization tunnel, compressing data as it travels from one BIG-IP system to the other across the WAN.

**RAM caching**

The BIG-IP system can store HTTP objects in the BIG-IP system’s RAM. Subsequent connections can then re-use these objects, to reduce the amount of load on back-end servers.

**Rate shaping**

*Rate shaping* is a feature that allows you to categorize certain types of connections into rate classes, for the purpose of customizing the throughput of those connections. This is useful, for example, when you want to optimize web-server performance for preferred Internet customers.
TCP optimizations

The BIG-IP system includes significant TCP optimizations, such as in-order delivery and content spooling.

Enhancing network security

Security is an important consideration in managing local network traffic. Accordingly, the BIG-IP system contains a number of features designed to assist in preventing security breaches. These features pertain not only to authenticating and authorizing users and applications, but also to detecting intrusions and mitigating DOS attacks.

In general, when the BIG-IP system detects a security problem, it can take actions such as these:

- Reject a client request based on SSL certificate verification
- Reject and discard unauthorized packets
- Alert system administrators to an attack or infiltration attempt
- Direct suspicious traffic to specific target servers
- Log authentication failures
- Prevent SYN flooding

An important consideration for any networked environment is the authentication and authorization mechanism that you use to authenticate users and their client requests and to control user and application access to server resources. To this end, the BIG-IP system supports Pluggable Authentication Module (PAM) technology, and provides a complete set of PAM authentication modules that you can choose from to handle your authentication or authorization needs.

The authentication modules that the BIG-IP system provides are as follows:

- **An LDAP module**
  Uses a remote LDAP server to perform user name/password user authentication.

- **A RADIUS module**
  Uses a Remote Authentication Dial In User Service (RADIUS) server to perform user name/password user authentication.

- **A TACACS+ module**
  Uses a remote Terminal Access Controller Access Control System (TACACS+) server to perform user name/password user authentication.

- **An SSL Client Certificate LDAP module**
  Uses a remote LDAP server to perform SSL certificate-based authorization of client SSL traffic.

- **An OCSP module**
  Uses a remote Online Certificate Status Protocol (OCSP) server to provide up-to-date SSL certificate revocation status for the purpose of authenticating client and server SSL traffic.
Chapter 1

- **A CRLDP module**
  Uses the industry-standard Certificate Revocation List Distribution Point (CRLDP) technology to manage SSL certificate revocation status for the purpose of authenticating client and server SSL traffic.

- **A Kerberos Delegation module**
  Authenticated client traffic with Microsoft Windows Integrated Authentication.
Introducing Local Traffic Management

Summarizing local traffic management configuration

Once you have set up your base network and you have administrative access to the BIG-IP system, and at least a default VLAN assignment for each interface, the next step is to configure a network for managing traffic targeted to your internal servers. For more information, see the BIG-IP Systems: Getting Started Guide.

At the heart of the BIG-IP system are virtual servers and load balancing pools. Virtual servers receive incoming traffic, perform basic source IP and destination IP address translation, and direct traffic to servers, which are grouped together in load balancing pools.

Using the Configuration utility

All users need to use the web-based Configuration utility in order to license the system for the first time.

In addition to using the Setup utility to set up the management network and initial traffic management software configuration, you use the Configuration utility to customize and maintain the BIG-IP system. In the Configuration utility, you can also monitor current system performance, and view a network map that shows the virtual servers that you have created, along with the pools (and pool members) that the virtual servers reference.

For information on setting user preferences for the Configuration utility, see the TMOS Management Guide for BIG-IP Systems. For information on supported browsers, see the applicable release notes on the Ask F5 Knowledge Base web site, http://support.f5.com.

Configuring local traffic objects

To configure a basic local traffic management system, you use the Configuration utility. With this utility, you can create a complete set of configuration objects that work together to perform local traffic management. Each object has a set of configuration settings that you can use as is or change to suit your needs.

To configure local traffic objects, you can configure each type of object individually, or, in some cases, you can use the application templates feature to configure all required objects in a single operation. For more information on using application templates, see Understanding application templates, on page 1-19.

The local traffic objects that you can configure are:

- **Virtual servers**
  The primary function of a virtual server is to receive requests and distribute them to pool members according to criteria you specify.

- **Application profiles**
  Application-type profiles contain settings that define the behavior of various application types that use protocols such as FTP and HTTP.
• **Session persistence profiles**
  Session persistence profiles allow you to implement session persistence based on a variety of criteria such as HTTP cookies, source IP addresses, and destination IP addresses.

• **Protocol profiles**
  Protocol profiles contain settings that define the behavior of Layer 4 traffic such as TCP, UDP, and SCTP traffic.

• **SSL profiles**
  SSL profiles enable the BIG-IP system to encrypt and decrypt both client-side and server-side data that passes through the BIG-IP system.

• **Authentication profiles**
  Authentication-type profiles provide authentication interoperability between remote authentication servers and the BIG-IP system.

• **iRules**
  iRules can define criteria for pool-member selection, as well as perform content transformations, logging, custom protocol support, and so on. For complete information on iRules, see the F5 Networks DevCentral web site, [http://devcentral.f5.com](http://devcentral.f5.com).

• **Load balancing pools**
  Load balancing pools contain servers to which requests can be sent for processing.

• **Nodes**
  Nodes represent server IP addresses on your network that you can enable and disable, and for which you can obtain status.

• **Monitors**
  Monitors track the current health or performance of pool members.

• **Rate Classes**
  Rate classes implement rate shaping, used to control bandwidth consumption.

• **Traffic Classes**
  Traffic classes regulate traffic flow based on a set of criteria that you define.

• **SNATs**
  Secure Network Address Translations (SNATs) translate the source IP address in a request, and allow multiple hosts to share the same source IP address.

• **SSL Certificates**
  The SSL Certificates object allows you to generate SSL certificate requests and install SSL certificates on the BIG-IP system, for the purpose of terminating and initiating SSL connections.

The three most basic objects in the BIG-IP system that you must configure for local traffic management are:

• Virtual servers
• Load balancing pools
• Profiles
Once you have configured local traffic objects on the system, either by configuring the objects individually or by using the using an application template, you can use the network map to display those objects. The network map displays a visual representation of the local traffic configuration that you have implemented, such as virtual servers and their associated pools. The network map also displays statistics about existing virtual servers, pools, nodes, and iRules™. For more information on the network map, see Using the network map feature, on page 1-14.

Managing timeout settings for connections and sessions

The BIG-IP system has a number of time-outs that can be set to promote active connection management. The system manages each load-balanced connection explicitly by keeping track of the connection in the connection table while the connection is still active. The connection table contains state information about client-side and server-side connections, as well as the relationships between them.

Each connection in the connection table consumes system resources to maintain the table entry and monitor connection status. The BIG-IP system must determine when a connection is no longer active and then retire the connection to avoid exhausting critical system resources. Resources such as memory and processor cycles are at risk if the connection table grows and remains unchecked.

You can also manage the duration of entries in the persistence table when using session persistence.

Connection reaping

Connections that close or reset in a normal way are retired from the connection table automatically. A significant number of connections, however, often remain idle without closing normally, for any number of reasons. Consequently, the BIG-IP system must reap these connections once they have been determined to be inactive. Reaping is the process of retiring or recycling connections that would otherwise remain idle.

To promote proactive reaping, you can configure several different timeout settings to tear down connections that have seen no active traffic after a specified period of time. While a few of these timeout settings are not user-configurable, you can actively configure most of these timeout settings, to meet the needs of any application.

Since you can configure timeout settings in multiple places, it is important to understand that sometimes more than one timeout setting affects the same connection. The optimal timeout configuration is one that retains idle connections for an appropriate amount of time (variable by application) before deciding that the connections are inactive and should be retired, to conserve system resources.
Idle timeout options

Idle connections can be timed out by protocol profiles or SNATs associated with the virtual server handling the connection. Connections that a virtual server does not manage can be timed out based on SNAT automap or VLAN group settings.

Table 1.1 shows a list of objects containing idle connection timeout settings that affect reaping. For each object type, the table lists the default value and whether that value is user-configurable.

<table>
<thead>
<tr>
<th>Configuration Object Type</th>
<th>Default in Seconds</th>
<th>User-configured?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast L4, Fast HTTP, TCP, and SCTP profiles</td>
<td>300</td>
<td>Yes</td>
</tr>
<tr>
<td>UDP profiles</td>
<td>60</td>
<td>Yes</td>
</tr>
<tr>
<td>SNAT automap</td>
<td>300</td>
<td>No</td>
</tr>
<tr>
<td>VLAN group</td>
<td>300</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1.1 Idle timeout settings that affect connection reaping

The shortest timeout value that applies to a connection is the value that always takes effect. In some cases, however, you might want to change this behavior.

For example, you might have configured a forwarding virtual server that is intended to carry long-standing connections, and these connections might become idle for long periods of time (such as SSH sessions). In this case, you can configure a long idle timeout value on the related protocol profile (in this case, TCP).

However, if the SNAT automap feature is also enabled, the default 300-second static timeout value still takes effect.
Other timeout settings

The BIG-IP system includes two other idle timeout settings, but these settings do not affect connection reaping. These settings appear in the OneConnect and persistence profile types. Table 1.2 shows the default values for these settings and whether the settings are user-configurable.

<table>
<thead>
<tr>
<th>Configuration Object Type</th>
<th>Default in Seconds</th>
<th>User-configured?</th>
</tr>
</thead>
<tbody>
<tr>
<td>OneConnect profiles</td>
<td>Disabled</td>
<td>Yes</td>
</tr>
<tr>
<td>Cookie Hash, Destination Address Affinity, Hash, SIP, Source Address Affinity, and Universal persistence profiles</td>
<td>180</td>
<td>Yes</td>
</tr>
<tr>
<td>MSRDP and SSL persistence profiles</td>
<td>300</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Table 1.2 Idle timeout settings that do not affect connection reaping*

The OneConnect timeout value controls the length of time that an idle server-side connection is available for re-use; that is, this timeout value might cause the system to close a server-side connection after becoming idle for a certain period of time. In this case, since that connection was never actively in use, no active client-side connections are affected, and the system transparently selects or establishes another server-side connection for new connections. The OneConnect timeout setting need not be coordinated with the idle timeout settings of other profiles.

Persistence timeout settings are actually idle timeout settings for a session, rather than for a single connection. Thus, persistence timeout settings should typically be set to a value slightly larger than the applicable connection idle timeout settings, to allow sessions to continue even if a connection within the session has expired.

Getting started

To configure a working traffic-management system, you need to create a virtual server, a load balancing pool, and possibly a profile.

Configuring virtual servers

When you create a virtual server, you specify the type of virtual server you want, that is, a host virtual server or a network virtual server. Then you can attach various properties and resources to it, such as application-specific profiles, session persistence, and user-written scripts called iRules that define pool-selection criteria. All of these properties and resources, when
associated with a virtual server, determine how the BIG-IP system manages local traffic. When you create and configure a virtual server, you use the Virtual Servers screen of the Configuration utility.

For more information on virtual servers, see Chapter 2, *Configuring Virtual Servers*.

### Configuring load balancing pools

A *load balancing pool* is a collection of internal servers that you group together to service client requests. A server in a pool is referred to as a *pool member*. Using the default load balancing algorithm, known as Round Robin, the BIG-IP system sends a client request to a member of that pool.

To implement a load balancing pool, you first create the pool, and then you associate the pool name with an existing virtual server. A virtual server sends client requests to the pool or pools that are associated with it.

Pools have settings associated with them, such as IP addresses for pool members, load balancing modes, and health and performance monitors. When you create a pool, you can use the default values for some of these settings, or change them to better suit your needs. To create and configure a load balancing pool, you use the Pools screen of the Configuration utility.

For more information on load balancing pools, see Chapter 4, *Configuring Load Balancing Pools*.

### Configuring profiles

A *profile* is a group of configuration settings that apply to a specific type of network traffic, such as HTTP connections. If you want the virtual server to manage a type of traffic, you can associate the applicable profile with the virtual server, and the virtual server applies that profile’s settings to all traffic of that type.

For example, you might want the BIG-IP system to compress HTTP response data. In this case, you can configure an HTTP profile to enable compression, and associate the profile with a virtual server. Then, when the virtual server processes an HTTP request, the BIG-IP system compresses the response.

There are several types of profiles that you can create for your own needs. They are: HTTP, FTP, SIP, RTSP, Persistence, Fast L4, Fast HTTP, HTTP Class, TCP, UDP, SCTP, Client and Server SSL, Authentication, OneConnect, NTLM, Statistics, and Stream. When you create a profile, you can use the default values for the settings, or change them to better suit your needs. To create and configure a profile, you use one of the profiles screens of the Configuration utility.

For more information on configuring profiles, see Chapter 5, *Understanding Profiles*, and any of the following chapters:

- Chapter 6, *Managing Application Layer Traffic*
- Chapter 7, *Enabling Session Persistence*
- Chapter 8, *Managing Protocol Profiles*
• Chapter 9, *Managing SSL Traffic*
• Chapter 10, *Authenticating Application Traffic*
• Chapter 11, *Using Additional Profiles*
Chapter 1

Using the network map feature

The Configuration utility includes a feature known as the network map. The network map feature shows a summary of local traffic objects, as well as a visual map of the virtual servers, pools, and pool members on the BIG-IP system. For both the local traffic summary and the network map, you can customize the display using a search mechanism that filters the information that you want to display, according to criteria that you specify. The system highlights in blue all matches from a search operation.

Understanding the filtering mechanism

You can filter the results of the network map feature by using the Type and Status lists in the filter bar, as well as a Search box. With the Search box, you can optionally type a specific string. Figure 1.1 shows the filtering options on the Network Map screen.

![Figure 1.1 The filtering options on the Network Map screen](image)

When using the Search box, you can specify a text string that you want the system to use in a search operation. The default is asterisk (*). The settings of the Status and Type fields determine the scope of the search. The system uses the specified search string to filter the results that the system displays on the screen.

For example, if you constrain the search to include only unavailable nodes whose IP address includes 10.10, the operation returns those nodes, along with the members of the pool, the pool itself, the associated virtual server, and any iRules that you explicitly applied to that virtual server. The system sorts results alphabetically, by virtual server name.

The system supports searching on names, IP address, and IP address:port combinations, in both IPv4 and IPv6 address formats. The system processes the string as if an asterisk wildcard character surrounds the string. For example, you specify 10, the system effectively searches as if you had typed *10*. You can also specifically include the asterisk wildcard character. For example, you can use the following search strings: 10.10.10.*:80, 10.10.*, and *:80. If you specifically include a wildcard character, the system treats the string accordingly. For example, if you specify 10*, the system assumes you want to search for objects whose IP addresses begin with 10.
◆ Tip

Browsers have limits as to how much data they can render before they become sluggish and halt processing. Mapping large configurations might approach those limits; therefore, memory constraints might prevent the system from producing a network map of the whole configuration. If this might happen, the system posts an alert indicating that you can use the Network Map summary screen to determine the complexity of the configuration, which can give you an indication of the size of the resulting map. You can modify the search criteria to return fewer results, producing a map that does not encounter those limits.

Displaying a summary of local traffic objects

When you first open the Network Map screen, the screen displays a summary of local traffic objects. This summary includes the type of objects specified with the search mechanism, the number of each type of object, and, for each object type, the number of objects with a given status.

The summary displays data for these object types:

- Virtual servers
- Pools
- Pool members
- Nodes
- iRules

◆ Note

A local traffic summary includes only those objects that are referenced by a virtual server. For example, if you have configured a pool on the system but there is no virtual server that references that pool, the local traffic summary does not include that pool, its members, or the associated nodes in the summary.

Figure 1.2, on page 1-16, shows an example of a network map screen that summarizes local traffic objects on the system.
Figure 1.2 Example of a local traffic summary

For each object type listed in the summary, the system shows the number of objects with each type of status. Table 1.3 shows the various status indicators that the summary screen can display for a local traffic object.

<table>
<thead>
<tr>
<th>Status indicator</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Available]</td>
<td>The objects are enabled and available (able to receive traffic).</td>
</tr>
<tr>
<td>![Unavailable]</td>
<td>The objects are enabled but are currently unavailable. However, the objects might become available later, with no user action required. An example of an object showing this status is a virtual server whose connection limit has been exceeded. When the number of connections falls below the configured limit, the virtual server becomes available again.</td>
</tr>
<tr>
<td>![Offline]</td>
<td>The objects are enabled but offline because an associated object has marked the object as unavailable. To change the status so that the object can receive traffic, you must actively enable the object.</td>
</tr>
<tr>
<td>![Unknown]</td>
<td>The status of the objects is unknown.</td>
</tr>
</tbody>
</table>

Table 1.3 Explanation of status icons in a local traffic summary
To display a summary of local traffic objects

1. On the Main tab of the navigation pane, expand Local Traffic, and click Network Map.
   The Network Map screen opens.

2. From the Status list, select a status.
   This causes the system to limit the search results to objects with that status. Possible values are Any Status (the default value), Available, Unavailable, Offline, and Unknown.

3. From the Type list, select an object type.
   This causes the system to limit the search results to objects of that type. Possible values are All Types (the default value), Virtual Servers, iRules, Pools, Pool Members, and Nodes.

4. If you want to further limit the search results, type a string in the Search box.
   For example, you can limit the search to only those objects that include the string 10.10 in their names.
   
   Note: For performance reasons, the system does not normally search within iRule text for the specified search string. If you want the search results to include iRules that contain the specified string, see step 5. Otherwise, see step 6.

5. To direct the system to search all iRule text, check the Search iRule Definition box.
   
   Note: Enabling this setting could affect system performance while the system performs the search operation.

6. Click the Update Summary button.
   This action refreshes the local traffic summary displayed on the screen.

Displaying the network map

The network map presents a visual hierarchy of the names and status of virtual servers, pools, pool members, nodes, and iRules defined on the system. The map shows all objects in context, starting with the virtual servers at the top. The Status, Type, and Search settings at the top of the screen determine the objects that the map includes.

When you position the cursor over an object on the map, the system presents hover text containing information about the object. When you position the cursor over the status icon accompanying an object, the system presents hover text containing information about the object’s status, text which also appears on the pool’s Properties screen.

The system arranges objects in alphabetic order, and organizes the dependent objects in a hierarchy.
Due to the way that a network map presents objects in context, the updated screen also shows objects of other statuses, types, and names that relate to those objects. This is because a network map always shows objects in context with the objects that depend on them, and the objects they depend on.

For example, if you have an available virtual server with an available pool and two pool members, one available and one offline, then selecting Offline from the Status list causes the system to show the offline pool member in context with the available virtual server and the available pool. This is because the available virtual server and the available pool depend on the offline pool member.

To display a network map

1. On the main tab of the navigation pane, expand Local Traffic, and click Network Map. The Network Map screen opens.
2. From the Status list, select a status. This causes the system to limit the search results to objects with that status. Possible values are Any Status (the default value), Available,Unavailable, Offline, and Unknown.
3. From the Type list, select an object type. This causes the system to limit the search results to objects of that type.
4. If you want to further limit the search results, type a string in the Search box. For example, you can limit the search to only those objects that include the string 10.10 in their names.
   
   Note: For performance reasons, the system does not normally search within iRule text for the specified search string. If you want the search results to include iRules that contain the specified string, see step 5. Otherwise, see step 6.
5. To direct the system to search all iRule text, check the Search iRule Definition box.
   
   Note: Enabling this setting could affect system performance while the system performs the search operation.
6. Click the Show Map button. This action displays the requested network map on the screen.
Understanding application templates

An application template is a wizard-like feature that queries the user for information related to a specific application. The BIG-IP system then automatically configures the BIG-IP system to process traffic for that application, based on that information.

An example of an application template is the Microsoft IIS template, which creates BIG-IP objects for managing traffic destined for a Microsoft IIS server.

Several application templates are available for you to use:

- BEA WebLogic
- Microsoft Exchange Outlook Web Access
- Microsoft IIS
- Microsoft SharePoint
- Oracle Application Server
- SAP ERP Central Component
- SAP Enterprise Portal
- VMware VDI

Benefits of application templates

Application templates ease the process of configuring the BIG-IP system. Instead of having to individually create each object that pertains to the type of application traffic you want the BIG-IP system to manage, you can run an application template. The application template automatically creates BIG-IP system objects that are customized for that application. These objects can be either local traffic objects, TMOS objects, or both, depending on the template you are using.

For example, when you run the Microsoft IIS application template, the BIG-IP system uses the information you provide to create a virtual server, pool, HTTP profile, and health monitor, all tailored to control Web traffic destined for an IIS server.

Working with application templates

Application templates are available for you to run from within the Configuration utility.

◆ Important

All local traffic objects that an application template creates reside in administrative partition Common. Consequently, to use the application templates feature, including viewing the Templates list screen, you must have a user role assigned to your user account that allows you to view and manage objects in partition Common.
To see all available templates, you can display the Templates list screen. Figure 1.3 shows an example of the Templates list screen in the Configuration utility.

Figure 1.3 Sample Templates list screen

The remainder of the Understanding application templates section provides generic information about the application templates feature. For application-specific deployment information, visit the web site www.f5.com and on the menu bar, click Solutions. From there, you can access the following deployment guides:

- Deploying the BIG-IP System with BEA WebLogic Server
- Deploying the BIG-IP System with Microsoft Internet Information Services (IIS) 7.0
- Deploying F5 with Microsoft Office SharePoint 2007
- Deploying F5 with Oracle Application Server 10g
- Deploying F5 with SAP NetWeaver and Enterprise SOA
- Deploying F5 with VMware Virtual Desktop Infrastructure (VDI)

Using an application template

You use an application template by setting the current administrative partition to Common and running the template from within the Configuration utility.

All available templates appear in the list of templates (shown in Figure 1.3 on this page). You then click the name of a template and provide information on the resulting screen.
To use an application template

1. Verify that your current administrative partition is set to Common.

2. On the Main tab of the navigation pane, expand Local Traffic, and click Templates and Wizards. The Templates screen opens, displaying a list of templates.

3. In the Application column, click a template name. The screen for that application opens.

4. Configure all settings on the screen.

5. Click Finished. A list of all objects that the template created displays on the screen.

Note that for any given application template, such as Microsoft IIS, you can create multiple deployments of that template. For example, using the Microsoft IIS template, you can create a separate deployment for each of three virtual servers, naming the deployments my_iis1, my_iis2, and my_iis3.

Sharing HTTP profiles

To operate as efficiently as possible, the BIG-IP system allows multiple deployments of an application template to share certain HTTP profiles that the template might create. This avoids multiple deployments of a template having to create and maintain identical sets of HTTP profiles.

Most application templates create sharable HTTP profiles. The indication that an HTTP profile is sharable by multiple deployments is the profile name, which is appended with the string _shared_http.

For example, when you create the first deployment of the Microsoft IIS template, the template might create a sharable HTTP profile named microsoft_iis_http_acceleration_shared_http. Thereafter, any subsequent deployment of the Microsoft IIS template can re-use that HTTP profile instead of creating a separate but identical HTTP profile.

Viewing application template objects

Once you have finished using the template, each local traffic object that the template created appears on the list screen for that object type. For example, if the application template created a load balancing pool, the pool then appears on the Pools List screen in the Configuration utility, along with all other pools that you are allowed to view on the BIG-IP system.

Identifying the objects that are associated with an application is simple, because each object that the template creates appends a user-specified prefix to the object name. For example, if you create a Microsoft IIS deployment, specifying the prefix IIS, each object that the template subsequently creates includes the prefix IIS in the name. For example, a pool that the template creates might be named IIS_pool, and a virtual server might be named IIS_virtual_server.
Deleting application template objects

After using a particular application template, if you want to delete all objects that the template created, you must access each relevant list screen in the Configuration utility and delete the application object on that screen. Continuing with the previous example, to delete the pool that the Microsoft IIS template created, you display the Pools list screen and delete the object IIS_pool. Similarly, to delete the virtual server, you display the Virtual Servers list screen and delete the object IIS_virtual_server. You continue this process until all relevant objects are deleted from the system.

Tip

You can use the Network Map feature within the Configuration utility to more easily identify template-related objects.
About this guide

Before you use this guide, we recommend that you run the Setup utility on the BIG-IP system to configure basic network and network elements such as static and floating self IP addresses, interfaces, and VLANs, to name a few. For more information, see the BIG-IP® Systems: Getting Started Guide.

After running the Setup utility, you can further customize your system by using the Configuration utility to create local traffic management objects such as virtual servers, load balancing pools, and profiles.

Finally, you can adjust the elements you have configured or add additional ones as your needs change.

Before you continue with adjusting or customizing your BIG-IP system configuration, complete these tasks:

- Choose a configuration tool.
- Familiarize yourself with additional resources such as product guides and online help.
- Review the stylistic conventions that appear in this chapter.

Finding additional information

In addition to this guide, there are other sources of the documentation you can use in order to work with the BIG-IP system. The following documentation pertains to the Local Traffic Manager product and is available in PDF format from the Ask F5® Knowledge Base website, http://support.f5.com. These guides are also available from the first web page you see when you access the browser-based Configuration utility:

- **BIG-IP® Systems: Getting Started Guide**
  This guide contains any information you need to initially install, license, and set up your BIG-IP system.

- **TMOS™ Management Guide for BIG-IP® Systems**
  This guide contains any information you need to configure and maintain the network and system-related components of the BIG-IP system. With this guide, you can perform tasks such as configuring routes and VLANs, assigning self IP addresses, creating administrative user accounts, and managing a redundant system.

- **BIG-IP® Local Traffic Manager: Implementations**
  This guide contains complete procedures for implementing specific goals, such as processing SSL traffic with data compression, or assigning privileges to remotely-authenticated user accounts. This guide ties together the detailed information contained in the Configuration Guide for BIG-IP® Local Traffic Management to help you implement specific traffic-management configurations.
Stylistic conventions

To help you easily identify and understand important information, our documentation uses the stylistic conventions described below.

Using the examples

All examples in this documentation use only private class IP addresses. When you set up the configurations we describe, you must use valid IP addresses suitable to your own network in place of our sample addresses.

Identifying new terms

To help you identify sections where a term is defined, the term itself is shown in bold italic text. For example, a virtual server is a specific combination of a virtual address and virtual port, associated with a content site that is managed by a BIG-IP system or other type of host server.

Identifying references to objects, names, and commands

We apply bold text to a variety of items to help you easily pick them out of a block of text. These items include web addresses, IP addresses, utility names, and portions of commands, such as variables and keywords. For example, you can set the Idle Timeout value to 5.

Identifying references to other documents

We use italic text to denote a reference to another document or section of a document. We use bold, italic text to denote a reference to a book title. For example, for installation instructions, see the guide titled BIG-IP® Systems: Getting Started Guide.

Identifying command syntax

We show complete commands in bold Courier text. Note that we do not include the corresponding screen prompt, unless the command is shown in a figure that depicts an entire command line screen. For example, the following command shows the configuration of the specified pool name:

```
bigpipe self <ip_address> show
or
b self <ip_Address> show
```
Table 1.4 explains additional special conventions used in command line syntax.

<table>
<thead>
<tr>
<th>Item in text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>Indicates that the command continues on the following line, and that users should type the entire command without typing a line break.</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Identifies a user-defined parameter. For example, if the command has &lt;your name&gt;, type in your name, but do not include the brackets.</td>
</tr>
<tr>
<td></td>
<td>Separates parts of a command.</td>
</tr>
<tr>
<td>[]</td>
<td>Indicates that syntax inside the brackets is optional.</td>
</tr>
<tr>
<td>...</td>
<td>Indicates that you can type a series of items.</td>
</tr>
</tbody>
</table>

*Table 1.4 Command line syntax conventions*
Finding help and technical support resources

You can find additional technical documentation and product information in the following locations:

◆ **Online help for local traffic management**
The Configuration utility has online help for each screen. The online help contains descriptions of each control and setting on the screen. Click the Help tab in the left navigation pane to view the online help for a screen.

◆ **Welcome screen in the Configuration utility**
The Welcome screen in the Configuration utility contains links to many useful web sites and resources, including:
  * The F5 Networks Technical Support web site
  * The F5 Solution Center
  * The F5 DevCentral web site
  * Plug-ins, SNMP MIBs, and SSH clients

◆ **F5 Networks Technical Support web site**
The F5 Networks Technical Support web site, [http://support.f5.com](http://support.f5.com), provides the latest documentation for the product, including:
  * Release notes for the BIG-IP system, current and past
  * Updates for guides (in PDF form)
  * Technical notes
  * Answers to frequently asked questions
  * The Ask F5® Knowledge Base

To access this site, you need to register at [http://support.f5.com](http://support.f5.com).
2

Configuring Virtual Servers

• Introducing virtual servers and virtual addresses
• Understanding virtual server types
• Understanding clustered multi-processing
• 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. 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 on the BIG-IP system that is represented by an IP address and a service. Clients on an external network can send application traffic to a virtual server, which then directs the traffic according to your configuration instructions. The main purpose of a virtual server is often to balance traffic load across a pool of servers on an internal network. 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 example, a virtual server can enable compression on HTTP request data as it passes through the BIG-IP system, or decrypt and re-encrypt SSL connections and verify SSL certificates. For each type of traffic, such as TCP, UDP, HTTP, SSL, SIP, 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 a specific traffic type. Through a virtual server, you can set up session persistence for HTTP, SSL, SIP, and MSRDP sessions, 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 a specific type of traffic
- Enable persistence for a specific type of traffic
- 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. For more information, see the chapter that describes VLANs and VLAN groups in the *TMOS™ Management Guide for BIG-IP® Systems*.

◆ **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 server.

◆ **Increasing the speed of processing HTTP traffic**
A *Performance (HTTP)* virtual server is a virtual server with which you associate a Fast HTTP profile. Together, the virtual server and profile increase the speed at which the virtual server processes HTTP requests.

◆ **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 2-5, 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 a virtual address. 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.

How does a virtual server work?

A good way to illustrate a virtual server is to examine the situation where a user on a client on an external network sends HTTP traffic through a Web browser. In this case, the BIG-IP system follows this process:

1. The user initiates a connection by entering a URL into a Web browser. The browser resolves the URL to a virtual server address that you have previously created on the BIG-IP system. This virtual server address is the destination address in the request.

2. The BIG-IP system examines the corresponding virtual server configuration and determines the pool of Web servers to which to send the incoming traffic.

3. The BIG-IP system examines the pool configuration to determine the load balancing algorithm to use to select an internal server node.

4. The BIG-IP system uses this algorithm to select the specific server node.

5. In the Destination IP Address header of the request packets, the BIG-IP system changes the destination IP address (that is, the virtual server address) to the actual address of the selected node. In this case, the source address in the packets (that is, the address of the client that initiated the connection) remains unchanged.

6. The BIG-IP system sends the incoming packets to the selected server node.

7. When the server node sends its response back to the client, the response returns through the BIG-IP system and a reverse translation occurs. In the Source IP Address header of the response packets, the BIG-IP system translates the actual source IP address of the response (the server node address) to the virtual server address. This causes the source address in the response to match the destination address in the request, a requirement necessary to ensure that the client accepts the response.
In short, the BIG-IP system typically translates the *destination address in a request* to the actual address of a server node, and translates the actual *source address in a response* to the virtual server address. This means that a client on an external network never sees the private class IP address of an internal server node.

**Note**

To ensure that a server response returns through the BIG-IP system, you must configure the default route on the server to be an internal VLAN’s self IP address. If you cannot do this because the server is on a different network than the BIG-IP system, you can create a SNAT instead. For more information, see Chapter 14, *Configuring SNATs.*
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 receiving a connection request for that site, the BIG-IP system 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 2-12.
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 2-13.

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.
Understanding clustered multi-processing

The BIG-IP system includes a performance feature known as clustered multi-processing, or CMP. **CMP** is a traffic acceleration feature that creates a separate instance of the Traffic Management Microkernel (TMM) service for each central processing unit (CPU) on the system. When CMP is enabled, the workload is shared equally among all CPUs.

Whenever you create a virtual server, the BIG-IP system automatically enables the CMP feature. When CMP is enabled, all instances of the TMM service process application traffic. (For information on configuring a virtual server, see *Creating a virtual server*, on page 2-11.)

Seeing differences in system behavior when CMP is enabled

Although the CMP feature is primarily an internal performance feature designed to accelerate application traffic, there are cases when you can see the effect of the feature on the system. Figure 2.1 shows output from the `bigpipe virtual show all` command, which shows that CMP is enabled for the virtual server **my_virtual**.

![Figure 2.1 bigpipe output showing CMP enabled for a virtual server](image)

When you view standard performance graphs using the Configuration utility, you can see multiple instances of the TMM service (**tmm0**, **tmm1**, and so on). Figure 2.2, on page 2-9, shows performance graphs that display performance for multiple CPUs and multiple TMM instances.
When CMP is enabled, be aware of the following facts:

- While displaying some statistics individually for each TMM instance, the BIG-IP system displays other statistics as the combined total of all TMM instances.
- Connection limits for a virtual server with CMP enabled are distributed evenly across all instances of the TMM service.
- Certain load balancing modes might behave differently than on non-CMP systems.

**Note**

We recommend that you disable the CMP feature if you set a small connection limit on pool members (for example, a connection limit of 2 for the 8400 platform or 4 for the 8800 platform).

### Disabling and re-enabling CMP

In some cases, you might want the TMM service to be assigned to a specific CPU. To do this, you can disable CMP altogether, although only on certain hardware platforms. You can disable CMP by configuring the bigdb variable `provision.tmmcount`, and then running the `bigstart restart` command:

```
bigpipe db provision.tmmcount 1
bigstart restart
```
To re-enable CMP, simply change the value of the bigdb variable from 1 to 0, and re-rerun the `bigstart restart` command, as follows:

```
bigpipe db provision.tmmcount 0
bigstart restart
```

**Understanding CMP on redundant systems**

If the CMP feature is enabled on a unit of a redundant system, note the following system behaviors:

- Each instance of the TMM service has a separate entry in the high availability (HA) table.
- The active unit fails over if any one instance of the TMM service fails.
- A VLAN fail-safe triggers failover only if all instances of the TMM service agree.
- Connection mirroring might be supported, depending on the hardware platform.
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 2-14. For information on viewing existing virtual server configurations, see *Managing virtual servers and virtual addresses*, on page 2-22.

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 2-5, and *Network virtual servers*, on page 2-5.) 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**

When you create a virtual server, the BIG-IP system places the virtual server into your current administrative partition. For information on partitions, see the *TMOSTM Management Guide for BIG-IP® Systems*.

**To create a virtual server**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**. The Virtual Servers screen opens.
2. On the upper right portion of the screen, click the **Create** button. The New Virtual Server screen opens.
   
   **Note:** If the Create button is unavailable, this indicates that your user role does not grant you permission to create a virtual server.
3. Configure the required settings. For more information, see Table 2.1, on page 2-14.
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 2-14.
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. For more information, see the TMOSTM Management Guide for BIG-IP® Systems.

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 2-6.

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 4, Configuring Load Balancing Pools.

To create a default wildcard virtual server

1. On the main tab of the navigation pane, expand Local Traffic, and click Virtual Servers.
2. On the upper right portion of the screen, click the Create button. The New Virtual Server screen opens.
   
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a virtual server.

3. Configure all required settings.
   
   Note: If the wildcard virtual server is to pertain to a route domain other than route domain 0, you must append the route domain ID to the destination address, using the %<ID> notation. For more information, see the TMOS® Management Guide for BIG-IP Systems.

4. Click Finished.
To create a port-specific wildcard virtual server

1. On the main tab of the navigation pane, expand Local Traffic, and click Virtual Servers. The Virtual Servers screen opens.


   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a virtual server.

3. In the Address box, type the wildcard IP address 0.0.0.0.

   Note: If the wildcard virtual server is to pertain to a route domain other than route domain 0, you must append the route domain ID to the destination address, using the %<ID> notation. For more information, see the TMOS® Management Guide for BIG-IP Systems.

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. For the Default Pool setting 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 of the navigation pane, expand Local Traffic, and click Virtual Servers. The Virtual Servers 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. In the Enable Translation section, verify that the Port box is cleared.
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 2-11.

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 2.1 lists and describes these general properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name that you assign to the virtual server. This property is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Destination Type</td>
<td>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 Understanding virtual server types, on page 2-5. This property is required.</td>
<td>Host</td>
</tr>
<tr>
<td>Destination Address</td>
<td>The IP address of the virtual server. Note that if you want the virtual server to pertain to a route domain other than the default route domain (route domain 0), you must use the %&lt;ID&gt; notation to append the route domain ID to the destination address. For example, if the destination address is 192.168.20.10 and you want the virtual server to pertain to route domain 2, you must specify the address as 192.168.20.10%2.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

Table 2.1 General properties of a virtual server
Configuring Virtual Servers

When creating a virtual server, you can configure a number of settings. Table 2.2 lists and describes these virtual server configuration settings. Because all of these settings have default values, you are not required to change these settings.

### Table 2.1 General properties of a virtual server

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Mask</td>
<td>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.</td>
<td>No default value</td>
</tr>
<tr>
<td>Service Port</td>
<td>A service name or port number for which you want to direct traffic. This property is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>State</td>
<td>The state of the virtual server, that is, <strong>Enabled</strong> or <strong>Disabled</strong>. 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 <strong>down</strong> state. <strong>Note:</strong> If no VLAN is specified, then the <strong>Enabled</strong> or <strong>Disabled</strong> setting applies to all VLANs.</td>
<td><strong>Enabled</strong></td>
</tr>
</tbody>
</table>

### Table 2.2 Configuration settings for a virtual server

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of virtual server configuration. Choices are: <strong>Standard</strong>, <strong>IP Forwarding (IP)</strong>, <strong>Forwarding (Layer 2)</strong>, <strong>Performance (HTTP)</strong>, <strong>Performance (Layer 4)</strong>, and <strong>Reject</strong>. For more information, see <a href="#">Introducing virtual servers and virtual addresses</a>, on page 2-1. Note that if set to <strong>Reject</strong>, this setting causes the BIG-IP system to reject any traffic destined for the virtual server IP address.</td>
<td><strong>Standard</strong></td>
</tr>
<tr>
<td>Protocol</td>
<td>The network protocol name for which you want the virtual server to direct traffic. Sample protocol names are <strong>TCP</strong> and <strong>UDP</strong>. 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. <strong>Note</strong> that this setting is disabled when creating a <strong>Performance (HTTP)</strong> type of virtual server.</td>
<td><strong>TCP</strong></td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Protocol Profile (Client)</td>
<td>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 8, Managing Protocol Profiles.</td>
<td>TCP</td>
</tr>
<tr>
<td>Protocol Profile (Server)</td>
<td>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 8, Managing Protocol Profiles.</td>
<td>(Use Client Profile)</td>
</tr>
<tr>
<td>OneConnect Profile</td>
<td>The name of an existing OneConnect™ profile for managing connection persistence. 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 8, Managing Protocol Profiles.</td>
<td>oneconnect</td>
</tr>
<tr>
<td>NTLM Conn Pool</td>
<td>The name of an existing NTLM profile. When used in conjunction with a OneConnect profile, an NTLM profile pools server-side connections for NT Lan Manager (NTLM) traffic.</td>
<td>None</td>
</tr>
<tr>
<td>HTTP Profile</td>
<td>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 6, Managing Application Layer Traffic.</td>
<td>None</td>
</tr>
<tr>
<td>FTP Profile</td>
<td>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 6, Managing Application Layer Traffic.</td>
<td>None</td>
</tr>
<tr>
<td>SSL Profile (Client)</td>
<td>The name of an existing SSL profile for managing client-side SSL 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 9, Managing SSL Traffic</td>
<td>None</td>
</tr>
<tr>
<td>SSL Profile (Server)</td>
<td>The name of an existing SSL profile for managing server-side SSL 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 9, Managing SSL Traffic.</td>
<td>None</td>
</tr>
<tr>
<td>Authentication Profile</td>
<td>The name of an existing authentication profile for managing an authentication mechanism. Examples are a remote LDAP or RADIUS server. 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 10, Authenticating Application Traffic.</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.2 Configuration settings for a virtual server
### Configuring Virtual Servers

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream Profile</td>
<td>The name of an existing Stream profile for searching and replacing strings within a data stream, such as a TCP connection. 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 11, Using Additional Profiles.</td>
<td>None</td>
</tr>
<tr>
<td>RTSP</td>
<td>The name of an existing RTSP profile. <strong>Real Time Streaming Protocol</strong> (RTSP) is a protocol used for streaming-media presentations. Using RTSP, a client system can control a remote streaming-media server and allow time-based access to files on a server.</td>
<td>None</td>
</tr>
<tr>
<td>SMTP</td>
<td>The name of an existing SMTP profile. SMTP profiles pertain to the Simple Mail Transport Protocol (SMTP). Note that this setting only appears when Protocol Security Module is licensed on the system. For more information, see the <strong>Configuration Guide for BIG-IP® Protocol Security Module</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>SIP Profile</td>
<td>The name of an existing SIP profile for managing SIP traffic. For more information, see <strong>Configuring SIP profile settings</strong>, on page 6-42. Please note that the SIP Profile option is only available with a Standard type virtual server.</td>
<td>None</td>
</tr>
<tr>
<td>Statistics Profile</td>
<td>The name of a statistics profile. For more information, see Chapter 11, Using Additional Profiles.</td>
<td>stats</td>
</tr>
<tr>
<td>VLAN Traffic</td>
<td>The names of VLANS for which the virtual server is enabled or disabled. For background information on VLANS, see the <strong>TMOS® Management Guide for BIG-IP® Systems</strong>.</td>
<td>ALL VLANS</td>
</tr>
<tr>
<td>Rate Class</td>
<td>The name of an existing rate class, used for enforcing a throughput policy for incoming network traffic. For more information, see Chapter 16, Configuring Rate Shaping.</td>
<td>None</td>
</tr>
<tr>
<td>Traffic Class</td>
<td>A list of the traffic classes you would like to assign to the virtual server. Any traffic flows that match the criteria defined in a traffic class are tagged with a classification ID. For more information, see Chapter 15, Configuring Traffic Classes.</td>
<td>No default value</td>
</tr>
<tr>
<td>Connection Limit</td>
<td>The maximum number of concurrent connections allowed for the virtual server. Setting this to 0 turns off connection limits.</td>
<td>0</td>
</tr>
<tr>
<td>Connection Mirroring</td>
<td>A setting that mirrors connections from the active unit to the standby unit of a redundant pair. This setting provides higher reliability, but might affect system performance.</td>
<td>Disabled (unchecked)</td>
</tr>
</tbody>
</table>

**Important:** To ensure that a standby unit retains its mirrored connections after a reboot operation, we recommend that you enable connection mirroring on Performance (Layer 4) virtual servers only. For more information, see the description in this table of the **Type** setting, as well as What is a virtual server?, on page 2-1. We also recommend that you set up a direct link (trunk) between the peer units as a way to dedicate bandwidth for mirroring the connections. This prevents potential performance problems or loss of mirrored information.

---

**Table 2.2** Configuration settings for a virtual server
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Translation</td>
<td>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.</td>
<td>Enabled (checked)</td>
</tr>
<tr>
<td>Port Translation</td>
<td>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.</td>
<td>Enabled (checked)</td>
</tr>
</tbody>
</table>
| Source Port              | Specifies whether the system preserves the source port of the connection. Possible values are:  
  - **Preserve**: Specifies that the system preserves the value configured for the source port, unless the source port from a particular SNAT is already in use, in which case the system uses a different port.  
  - **Preserve Strict**: Specifies that the system preserves the value configured for the source port. If the port is in use by another connection, the system uses that source port anyway, and the destination server cannot distinguish the traffic of the connections sharing that source port. F5 Networks recommends that you restrict use of this setting to cases that meet at least one of the following conditions:  
    - The port is configured for UDP traffic.  
    - The system is configured for nPath routing or is running in transparent mode (that is, there is no translation of any other Layer 3 or Layer 4 field).  
    - There is a one-to-one relationship between virtual IP addresses and node addresses, or clustered multi-processing (CMP) is disabled.  
  - **Change**: Specifies that the system changes the source port. This setting is useful for obfuscating internal network addresses. | Preserve |
| SNAT Pool                | This setting assigns an existing SNAT pool to the virtual server, or enables the Automap feature. When you use this setting, the BIG-IP system automatically maps all original source IP addresses passing through the virtual server to an address in the SNAT pool. Possible values are: **None**, **Auto Map**, or the name of an existing SNAT pool. For more information, see Chapter 14, Configuring SNATs. | None |
| Clone Pool (Client)      | Used for intrusion detection, this feature causes the virtual server to replicate client-side traffic (prior to address translation), to a member of the specified clone pool. A clone pool receives all of the same traffic as the normal pool. You therefore use clone pools to copy traffic to intrusion detection systems. You can also configure the Clone Pool (Server) setting. | None |

*Table 2.2 Configuration settings for a virtual server*
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 2.3 lists and describes the specific resources that you can assign to a load balancing virtual server.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone Pool (Server)</td>
<td>Used for intrusion detection, this feature that causes the virtual server to replicate server-side traffic (after address translation), to a member of the specified clone pool. A clone pool receives all of the same traffic as the normal pool. You therefore use clone pools to copy traffic to intrusion detection systems. You can also configure the <strong>Clone Pool (Client)</strong> setting.</td>
<td></td>
</tr>
<tr>
<td>Last Hop Pool</td>
<td>A setting that directs reply traffic to the last hop router using a last hop pool. This overrides the <strong>auto_lasthop</strong> 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 <strong>auto_lasthop</strong> global variable is enabled, as it is by default. If you want to exclude one or more routers from <strong>auto_lasthop</strong>, or if the global <strong>auto_lasthop</strong> is disabled for any reason (for example, when you have an SSL gateway), you can use a last hop pool instead. (If <strong>auto_lasthop</strong> is enabled, the last hop pool takes precedence.) Before configuring a last hop pool, you must first create a pool containing the router inside addresses.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Table 2.2 Configuration settings for a virtual server**
Table 2.4 lists and describes the general properties and configuration settings of a virtual address.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>iRules</td>
<td>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. Similarly, if you are implementing an authentication iRule, you must configure a default or custom authentication profile. 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 17, Writing iRules.</td>
<td>No default value</td>
</tr>
<tr>
<td>HTTP Class Profiles</td>
<td>A list of existing HTTP Class profiles that you want the virtual server to use when load balancing its connections. For more information, see Chapter 8, Managing Protocol Profiles.</td>
<td>No default value</td>
</tr>
<tr>
<td>Default Pool</td>
<td>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 4, Configuring Load Balancing Pools.</td>
<td>No default value</td>
</tr>
<tr>
<td>Default Persistence Profile</td>
<td>The type of persistence that you want the BIG-IP system to use. This setting is available for Standard, Performance (HTTP), and Performance (Layer 4) types of virtual servers only. For more information, see Chapter 7, Enabling Session Persistence.</td>
<td>None</td>
</tr>
<tr>
<td>Fallback Persistence Profile</td>
<td>The type of persistence that the BIG-IP system should use if the system cannot use the specified default persistence. Valid types of persistence profiles for this setting are Source Address Affinity profiles and Destination Address Affinity profiles. You can also specify None. This setting is available for Standard, Performance (HTTP), and Performance (Layer 4) types of virtual servers only. For more information, see Chapter 7, Enabling Session Persistence.</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.3 Resources assigned to a load balancing virtual server

Configuring virtual address settings

The Configuration utility displays virtual address properties and settings. Table 2.4 lists and describes the general properties and configuration settings of a virtual address.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The IP address of the virtual server, not including the service.</td>
<td>No default value</td>
</tr>
<tr>
<td>Unit ID</td>
<td>The ID of the redundant-pair unit to which this address should apply.</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.4 General properties and configuration settings of a virtual address
## Configuring Virtual Servers

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>The availability of the virtual address with respect to service checking.</td>
<td>No default value</td>
</tr>
<tr>
<td>State</td>
<td>The state of the virtual address, that is, <strong>enabled</strong> or <strong>disabled</strong>.</td>
<td><strong>Enabled</strong></td>
</tr>
<tr>
<td>Advertise Route</td>
<td>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 <strong>Route Advertisement</strong> setting is enabled (checked). Possible values are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>When any virtual server is available</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>When all virtual server(s) are available</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Always</strong></td>
<td></td>
</tr>
<tr>
<td>Connection Limit</td>
<td>The number of concurrent connections that the BIG-IP system allows on this virtual address.</td>
<td>0</td>
</tr>
<tr>
<td>ARP</td>
<td>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.</td>
<td><strong>Enabled</strong> (checked)</td>
</tr>
<tr>
<td>Route Advertisement</td>
<td>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.</td>
<td><strong>Enabled</strong> (checked)</td>
</tr>
</tbody>
</table>

*Table 2.4* 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.

◆ Note

You can manage only those virtual servers and virtual addresses that you have permission to manage, based on your user role and partition access assignment. Only user accounts that are assigned the Administrator, Resource Administrator, or Manager role can manage virtual servers and virtual addresses.

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
- Partition in which the virtual server resides
- Destination (virtual address)
- Service port
- Type of virtual server
- Resources (associated pool, HTTP Class profile, iRules, and persistence profiles)
Note

If you are running Global Traffic Manager on the BIG-IP system, and you have configured a listener, the listener appears as a virtual server in the list of virtual servers within Local Traffic Manager.

To view a list of virtual servers

On the Main tab of the navigation pane, 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 of the navigation pane, expand Local Traffic, and click Virtual Servers.
The Virtual Servers screen opens.
2. In the Name column, click a virtual server name.
This displays the properties for that virtual server.
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 of the navigation pane, expand Local Traffic, and click Virtual Servers.
The Virtual Servers screen opens.
2. Click a virtual server name.
This displays the properties for that virtual server.
3. On the menu bar, click Resources.
The screen displays additional settings for the selected virtual server.
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**

2. In the Name column, click the name of a virtual server.
3. From the Statistics menu, choose Virtual Server. This displays the statistics for the virtual server.

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 of the navigation pane, expand Local Traffic, and click Virtual Servers. The Virtual Servers screen opens.
2. On the menu bar, click Virtual Address List. A list of existing virtual addresses appears.
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 of the navigation pane, expand Local Traffic, and click Virtual Servers. The Virtual Servers screen opens.
2. On the menu bar, click Virtual Address List. A list of existing virtual addresses appears.
3. In the Address column, click a virtual address. This displays the properties for that virtual address.
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 of the navigation pane, expand Local Traffic, and click Virtual Servers. The Virtual Servers screen opens.
2. On the menu bar, click Virtual Address List. A list of existing virtual addresses appears.
3. From the Statistics menu, choose Virtual Address. This displays statistics for the virtual address.

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 2.5. To display the icons within the Configuration utility, see To view or modify virtual server properties, on page 2-23, and To view or modify virtual address properties, on page 2-25.

<table>
<thead>
<tr>
<th>Status indicator</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Green Circle]</td>
<td>The virtual server or virtual address is enabled and able to receive traffic.</td>
</tr>
<tr>
<td>![Orange Exclamation]</td>
<td>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. 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.</td>
</tr>
<tr>
<td>![Red Up Arrow]</td>
<td>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.</td>
</tr>
<tr>
<td>![Black Square]</td>
<td>The virtual server or virtual address is operational but set to Disabled. To resume normal operation, you must manually enable the virtual server or virtual address.</td>
</tr>
<tr>
<td>![Blue Square]</td>
<td>The status of the virtual server or virtual address is unknown.</td>
</tr>
</tbody>
</table>

Table 2.5 Explanation of 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 of the navigation pane, expand Local Traffic, and click Virtual Servers.
   The Virtual Server 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 of the navigation pane, expand Local Traffic, and click Virtual Servers.
   The Virtual Server screen opens.
2. On the menu bar, click Virtual Address List.
   A list of existing virtual addresses appears.
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 of the navigation pane, expand Local Traffic, and click Virtual Servers.
   The Virtual Server screen opens.
2. Check the Select box to the left of the virtual server that you want to delete.
To delete a virtual address

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. This displays the Delete Confirmation screen.
5. Click Delete. This removes the virtual address.
3

Configuring Nodes

• Introducing nodes
• Creating and modifying nodes
• Configuring node settings
• Managing nodes
Introducing nodes

A **node** is a logical object on the BIG-IP® system that identifies the IP address of a physical resource on the network. 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 **10.10.10:80**, 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 4, 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 3-4, 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 Local Traffic Management.

◆ Important

When you create a node, the BIG-IP system places the node into your current administrative partition. For information on partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

To create a node

1. On the Main tab of the navigation pane, expand Local Traffic, and click Nodes.
   The Nodes screen opens.
2. In the upper-right corner of the screen, click Create.
   The New Node screen opens.
   
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a node.
3. For the Address setting:
   - If you want the node to pertain to the default route domain (route domain 0), type an IP address for the node.
• If you want the node to pertain to a route domain other than the default route domain (0), append the route domain ID to the IP address, as in this example: 10.12.10.1%2. In this example, the %2 notation designates that the IP address 10.12.10.1 pertains to route domain 2.

    Note: For more information on route domains, see the TMOSTM Management Guide for BIG-IP® Systems.

4. Specify, retain, or change each of the other settings.
5. Click Finished.

**To modify an existing node**

1. On the Main tab of the navigation pane, expand Local Traffic, and click Nodes.
   The Nodes screen opens.
2. In the Address column, click an address.
   This displays the settings for that node.
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 3.1 lists these configurable settings and their default values. Following this table are descriptions of specific settings.

<table>
<thead>
<tr>
<th>Node settings</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Specifies the IP address of the node. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Name</td>
<td>Specifies the name of the node.</td>
<td>No default value</td>
</tr>
<tr>
<td>Health Monitors</td>
<td>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.</td>
<td>Node default</td>
</tr>
<tr>
<td>Select Monitors</td>
<td>Specifies the monitors that the BIG-IP system is to associate with the node. This setting is only available when you set the Health Monitors setting to Node Specific.</td>
<td>No default value</td>
</tr>
<tr>
<td>Availability Requirement</td>
<td>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.</td>
<td>All</td>
</tr>
<tr>
<td>Ratio</td>
<td>Specifies the ratio weight you want to assign to the node.</td>
<td>1</td>
</tr>
<tr>
<td>Connection Limit</td>
<td>Specifies the maximum number of concurrent connections allowed on a node.</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.1 Node configuration settings

Before configuring a node, it is helpful to have a description of certain node settings that you might want to change.
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 10.12.10.1. This is the only required setting.

If you want the node to pertain to a route domain other than the default route domain (route domain 0), you must use the %<ID> notation to append the route domain ID to the node address. For example, if the node address is 10.12.10.1 and you want the node to pertain to route domain 2, you must specify the address as 10.12.10.1%2.

For more information on route domains, see the TMOSTM Management Guide for BIG-IP® Systems.

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.

◆ Note

Any monitor that you associate with a node must reside either in partition Common or in the partition that contains the node.

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.

◆ Note

If you use the bigpipe utility commands node and save to assign the same monitor to multiple nodes (for example, bigpipe node all monitor icmp and bigpipe save), the BIG-IP system creates a separate monitor-node entry in the bigip.conf file for each node.
For more information about health and performance monitors, see Chapter 12, 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 of the navigation pane, expand Local Traffic, and click Nodes. The Nodes screen opens.
2. On the menu bar, click Default Monitor.
3. In 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 Availability Requirement list, do one of the following:
   a) Select All.
      This specifies that all active monitors must succeed before the node is considered to be up.
   b) 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.
7. Click Update.

You should keep the following in mind when working with default monitors:

- If a user with permission to manage objects in partition Common disables a monitor that is designated as the default monitor for nodes (such as the icmp monitor), this affects all nodes on the system. Ensure that the default monitor for nodes always resides in partition Common.
- To specify default monitors, you must have the Administrator user role assigned to your user account.
If all nodes reside in the same partition, the default monitor must reside in that partition or in partition Common. If nodes reside in separate partitions, then the default monitor must reside in partition Common.

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 4, 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

◆ Note

You can manage only those nodes that you have permission to manage, based on your user role and partition access assignment.

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
• Partition in which the node resides
• 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 of the navigation pane, expand Local Traffic, and click Nodes.
This opens the Nodes screen and displays a list of nodes.

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.

◆ Partition
The partition in which the node resides.
◆ **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 of the navigation pane, expand Local Traffic, and click **Nodes**.
   The Nodes screen opens.
2. In the Address column, click the address of the node you want to view.
   This displays the settings for that node.

**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 3.2, on page 3-10. To display the icons within the Configuration utility, see **To view node properties**, following.

◆ **Tip**

*You can manually set the availability of a node by configuring the Manual Resume attribute of the associated health monitor. For more information, see Chapter 12, Configuring Monitors.*
<table>
<thead>
<tr>
<th>Status indicator</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Green Circle" /></td>
<td>The node is enabled and able to receive traffic.</td>
</tr>
<tr>
<td><img src="image" alt="Yellow Exclamation Mark" /></td>
<td>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 <strong>Connection Limit</strong> setting.</td>
</tr>
<tr>
<td><img src="image" alt="Red Diamond" /></td>
<td>The node is enabled but offline because an associated monitor has marked the node as <strong>down</strong>. To change the status so that the node can receive traffic, user intervention is required.</td>
</tr>
<tr>
<td><img src="image" alt="Black Circle" /></td>
<td>The node is set to <strong>Disabled</strong>, although a monitor has marked the node as <strong>up</strong>. To resume normal operation, you must manually enable the node.</td>
</tr>
<tr>
<td><img src="image" alt="Black Circle" /></td>
<td>The node is set to <strong>Disabled</strong> and is <strong>down</strong>. To resume normal operation, you must manually enable the node.</td>
</tr>
<tr>
<td><img src="image" alt="Black Diamond" /></td>
<td>The node is set to <strong>Disabled</strong> and is offline either because a user disabled it, or a monitor has marked the node as <strong>down</strong>. To resume normal operation, you must manually enable the node.</td>
</tr>
</tbody>
</table>
| ![Blue Square](image) | 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 3.2* **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 whatsoever.)

To enable or disable a node

1. On the Main tab of the navigation pane, expand Local Traffic, and click Nodes.
   This Nodes screen opens.
2. Locate the address of the node you want to enable or disable.
3. In the column to the left, check the Select box.
4. 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 of the navigation pane, expand Local Traffic, and click Nodes.
   The Nodes screen opens.
2. Locate the address of the node you want to enable or disable.
3. In the column to the left, check the Select box.
4. On the bottom of the screen, click Delete.
   A confirmation screen appears.
5. 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 3-5.
To remove an explicit monitor association for a node

1. On the Main tab of the navigation pane, expand Local Traffic, and click Nodes. The Nodes screen opens.
2. Click the address of the node you want to manage.
3. In the Configuration section of the screen, locate the Health Monitors setting.
4. Select Node Default or None.
5. Click Update.

To remove a default monitor

1. On the Main tab of the navigation pane, expand Local Traffic, and click Nodes. The Nodes screen opens.
2. Click the Default Monitor menu.
3. Using the Move button (>>), move any active monitors from the Active box to the Available box.
4. Click Update.
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® local traffic management 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 (and a service) on the network. A pool member can pertain to the default route domain on the BIG-IP system or to a specific route domain that you choose. 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 4-11.

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 Local Traffic Management.

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 and specify its members.
- Second, you must associate the pool with a virtual server.

Important

When you create a pool, the BIG-IP system places the pool into your current administrative partition. For information on partitions, see the TMOS Management Guide for BIG-IP Systems.

To create a load balancing pool

2. In the upper-right corner of the screen, click Create. The New Pool screen opens.
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a pool.
3. From the Configuration list, select Advanced.
4. For the Name setting, type a name for the pool.
5. Specify, retain, or change each of the other settings.
   For information on pool settings, see Configuring pool settings, on page 4-6, or refer to the online help for this screen.

6. Click Finished.

**To implement a load balancing pool**

1. On the Main tab of the navigation pane, expand Local Traffic, and click Virtual Servers.
   The Virtual Servers screen opens.
2. Click the name of the appropriate virtual server.
   This displays the settings for that virtual server.
3. On the menu bar, click Resources.
4. In 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 4-6, or see the online help. For information on adding members to an existing pool, see Modifying pool membership, on page 4-4.

**To modify pool settings**

1. On the Main tab of the navigation pane, expand Local Traffic, and click Pools.
   The Pools screen opens.
2. Click the name of an existing pool.
   This displays the existing settings for that pool.
3. From the Configuration list, select Advanced.
   This displays the pool settings.
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 of the navigation pane, expand Local Traffic, and click Pools.
   The Pools screen opens.
2. In the Members column, click the number shown.
   This lists the existing members of the pool.
3. Locate the Current Members section of the screen.
4. 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 4-16.
5. 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 4-20.
To add members to a load balancing pool

1. On the Main tab of the navigation pane, expand Local Traffic, and click Pools.
The Pools screen opens.

2. In the Members column, click the number shown.
   This lists the existing members of the pool.

3. In the right side of the screen, click Add.
The New Pool Members screen opens.

4. In the Address box:
   a) If the corresponding node address does not yet exist on the BIG-IP system, select New Address and type an IP address.
   b) If the corresponding node address already exists on the BIG-IP system, select Node List and select an existing node address from the list.

   Note: If you want the pool member to pertain to a route domain other than the default route domain (route domain 0), you must use the New Address box and append the route domain ID to the IP address of the pool member. For example, if the IP address you are specifying for the pool member is 10.10.10.4, and you want that member to pertain to route domain 2, you must append the route domain ID to the address using the %<ID> notation, as in this example: 10.10.10.4%2. For more information on route domains, see the TMOSTM Management Guide for BIG-IP® Systems.

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 4-16.

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 4-2.

Table 4.1 lists the settings that you can configure for a pool, followed by a description of each setting.

<table>
<thead>
<tr>
<th>Pool Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>You can specify the user-supplied name of the pool. Specifying a name for a pool is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Health Monitors</td>
<td>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.</td>
<td>No default value</td>
</tr>
<tr>
<td>Availability Requirement</td>
<td>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.</td>
<td>All</td>
</tr>
<tr>
<td>Allow SNAT</td>
<td>You can configure a pool so that SNATs are automatically enabled or disabled for any connections using that pool.</td>
<td>Yes</td>
</tr>
<tr>
<td>Allow NAT</td>
<td>You can configure a pool so that NATs are automatically enabled or disabled for any connections using that pool.</td>
<td>Yes</td>
</tr>
<tr>
<td>Action on Service Down</td>
<td>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.</td>
<td>None</td>
</tr>
<tr>
<td>Slow Ramp Time</td>
<td>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.</td>
<td>0</td>
</tr>
<tr>
<td>IP ToS to Client</td>
<td>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.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>IP ToS to Server</td>
<td>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.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>Link QoS to Client</td>
<td>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.</td>
<td>Pass Through</td>
</tr>
</tbody>
</table>

Table 4.1 Settings for a load balancing pool
Configuring Load Balancing Pools

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.

### 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

---

**Table 4.1 Settings for a load balancing pool**

<table>
<thead>
<tr>
<th>Pool Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link QoS to Server</td>
<td>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.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>Load Balancing Method</td>
<td>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.</td>
<td>Round Robin</td>
</tr>
<tr>
<td>Priority Group Activation</td>
<td>You can assign pool members to priority groups within the pool.</td>
<td>Disabled</td>
</tr>
<tr>
<td>New Members</td>
<td>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, a priority group, and a route domain.</td>
<td>No default value</td>
</tr>
</tbody>
</table>
not pools or pool members. That is, the destination address in the monitor specifies an IP address only, rather than an IP address and a service port. 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 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 12, *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 14, Configuring SNATs.

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 (TCP-only) or ICMP message.
- **Drop** -- The BIG-IP system simply cleans up the connection.
- **Reselect** -- The BIG-IP system selects a different node.

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 will be 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 B, Additional Monitor Considerations.

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

The Fastest methods pass a new connection based on the least number of outstanding Layer 7 requests to a pool member and the number of open Layer 4 connections. When the BIG-IP system receives a Layer 7 request, the BIG-IP system increments a counter for the pool member handling the connection. The BIG-IP decrements this counter when the corresponding Layer 7 response is received from the node. For requests presented to the node, this counter represents the current number of those requests that have not yet received a response. The Fastest 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).

The following rules apply to the Fastest load balancing methods:

- These methods require that you assign both a Layer 7 and a TCP type of profile to the virtual server.
- If a Layer 7 profile is not configured, the virtual server falls back to Least Connections load balancing mode.

- **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.

  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)**
  
  With the Observed methods, nodes are ranked based on the number of connections. Nodes that have a better balance of fewest connections receive a greater proportion of the connections.

  The Observed methods are different from the Least Connections method, in that the Least Connections method measures connections only at the moment of load balancing, while the Observed method tracks the number of Layer 4 connections to each node over time and creates a ratio for load balancing.

  The Observed modes work well in any environment, but may be particularly useful in environments where node performance varies significantly. These are dynamic load balancing methods.

  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 the number of current connections. 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. These are dynamic load balancing methods.
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 Sessions**
  The Least Sessions method passes a new connection to the node that currently has the least number of persistent sessions. Use of this load balancing method requires that the virtual server reference a type of persistence profile that tracks persistence connections. An example of this type of persistence profile is the Source Address Affinity or the Universal profile type. The Least Sessions method works best in environments where the servers or other equipment that you are load balancing have similar capabilities. This is a dynamic load balancing method.

- **L3 Address**
  The L3 Address method passes connections sequentially to each member, using its IP address. The IP address is a Layer three (L3) address.

### 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 4.1, on page 4-15 has three priority groups, 3, 2, and 1.
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 will 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. For the Address setting, you can click New Address and type an IP address (if the corresponding node address does not yet exist on the BIG-IP system), or you can click Node List and select an existing node IP address from the list.

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, following.

◆ Note

If you want the pool member to pertain to a route domain other than the default route domain (route domain 0), you must append the route domain ID to the IP address of the pool member. For example, if the IP address you are specifying for the pool member is 10.10.10.4, and you want that member to pertain to route domain 2, you must append the route domain ID to the address using the %ID notation, as in this example: 10.10.10.4%2. For more information on route domains, see the TMOS™ Management Guide for BIG-IP® Systems.
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 4-2. For information on adding members to an existing pool, see Adding members to an existing load balancing pool, on page 4-4.

Note

By design, a pool and its members always reside in the same partition.

Table 4.2 shows the settings that you can configure for an existing pool member, followed by a description of each setting.

<table>
<thead>
<tr>
<th>General property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio</td>
<td>Specifies the ratio weight that you want to assign to the pool member.</td>
<td>1</td>
</tr>
<tr>
<td>Priority Group</td>
<td>Specifies the priority group for the pool member.</td>
<td>1</td>
</tr>
<tr>
<td>Connection Limit</td>
<td>Specifies the maximum number of concurrent connections allowed for a pool member.</td>
<td>0</td>
</tr>
<tr>
<td>Health Monitors</td>
<td>Specifies whether you want the pool member to inherit the monitor associated with the pool or to use a different monitor.</td>
<td>Inherit From Pool</td>
</tr>
<tr>
<td>Select Monitors</td>
<td>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.</td>
<td>No default value</td>
</tr>
<tr>
<td>Availability Requirement</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2 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 4-24.

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 4-11, and Appendix B, 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 4-14.

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 12, *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 left arrows (**<<**), 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 circle-shaped 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 4-22, and *Understanding pool-member status*, on page 4-25.

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

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.

**Note**

You can manage only those pools that you have permission to manage, based on your user role and partition access assignment.
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
- Partition in which the pool resides
- 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 of the navigation pane, 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`.

- **Partition**
  The administrative partition in which the pool resides. For more information on administrative partitions, see the TMOS Management Guide for BIG-IP Systems.

- **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

2. Click a pool name. This displays the properties for that pool.
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 4.3. To locate the icons within the Configuration utility, see To view pool properties, on page 4-21. For background information on status icons, see Managing pools and pool members, on page 4-20.

<table>
<thead>
<tr>
<th>Status indicator</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>![green square]</td>
<td>At least one pool member is available for processing traffic.</td>
</tr>
<tr>
<td>![yellow triangle]</td>
<td>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.</td>
</tr>
<tr>
<td>![red diamond]</td>
<td>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.</td>
</tr>
<tr>
<td>![black square]</td>
<td>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: One or more pool members has no associated monitor. Monitor results are not available yet. The pool member’s IP address is misconfigured. The parent node has been disconnected from the network.</td>
</tr>
</tbody>
</table>

Table 4.3 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 4-4.

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

2. In the left column next to a pool name, check the Select box.
3. Click Delete. This displays the Delete Confirmation screen.
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.

◆ Note

You can manage only those pool members that you have permission to manage, based on your user role and partition access assignment.

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

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.

- **Partition**
  The partition in which the pool member resides. This partition always matches the partition of the pool itself.

- **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 of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools 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.

4. In the Current Members column, click a pool member IP address and port number.
   This displays the properties for that pool member.
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 4.4, on page 4-26. To view the icons within the Configuration utility, see To view pool-member properties, on page 4-24. For background information on status icons, see Managing pools and pool members, on page 4-20.

◆ Tip

You can manually set the availability of a pool member by configuring the Manual Resume attribute of the associated health monitor. For more information, see Chapter 12, Configuring Monitors.
Table 4.4 Explanation of status icons for pool members

<table>
<thead>
<tr>
<th>Status indicator</th>
<th>Explanation</th>
<th>State property is set to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Green Circle]</td>
<td>The pool member is set to <strong>Enabled</strong>, the parent node is <strong>up</strong>, and a monitor has marked the pool member as <strong>up</strong>.</td>
<td><strong>Enabled (All Traffic Allowed)</strong></td>
</tr>
<tr>
<td>![Yellow Triangle]</td>
<td>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 <strong>Connection Limit</strong> setting.</td>
<td><strong>Enabled (All Traffic Allowed)</strong></td>
</tr>
<tr>
<td>![Red Diamond]</td>
<td>The pool member is unavailable because either the parent node is down, a monitor has marked the pool member as <strong>down</strong>, or a user has disabled the pool member.</td>
<td><strong>Enabled (All Traffic Allowed)</strong></td>
</tr>
<tr>
<td>![Black Circle]</td>
<td>The pool member is set to <strong>Disabled</strong>, although a monitor has marked the pool member as <strong>up</strong>. To resume normal operation, you must manually enable the pool member.</td>
<td><strong>Disabled (Only persistent or active connections allowed)</strong></td>
</tr>
<tr>
<td>![Red Diamond]</td>
<td>The pool member is set to <strong>Disabled</strong> and is offline because the parent node is <strong>down</strong>. To resume normal operation, you must manually enable the pool member.</td>
<td><strong>Forced Offline (Only active connections allowed)</strong></td>
</tr>
<tr>
<td>![Red Diamond]</td>
<td>The pool member is set to <strong>Disabled</strong> and is offline because a user disabled it. To resume normal operation, you must manually enable the pool member.</td>
<td><strong>Disabled (Only persistent or active connections allowed)</strong></td>
</tr>
<tr>
<td>![Red Diamond]</td>
<td>The pool member is set to <strong>Disabled</strong> and is offline because either the parent node is <strong>down</strong>, or a monitor has marked the pool member as <strong>down</strong>. To resume normal operation, you must manually enable the pool member.</td>
<td><strong>Forced Offline (Only active connections allowed)</strong></td>
</tr>
<tr>
<td>![Black Square]</td>
<td>The pool member or node has no monitor associated with it, or no monitor results are available yet</td>
<td><strong>Enabled (All Traffic Allowed)</strong></td>
</tr>
</tbody>
</table>

◆ Note

Black status icons indicate that the **Manual Resume** attribute is enabled on the associated monitor. For more information, see Chapter 12, **Configuring Monitors**.
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 4-25.

**To enable or disable a pool member**

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

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 4-4.

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 of the navigation pane, expand Local Traffic, and click Pools.
   The Pools screen opens.
2. In the Name column, click a pool name.
   This displays the properties of the pool.
3. On the menu bar, click Members.
   This displays the list of pool members.

4. In the left column next to a pool member address, check the Select box.

5. Click Delete.
   This displays the Delete Confirmation screen.

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)
Understanding Profiles

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

The BIG-IP® local traffic management system can manage application-specific network traffic in a variety of ways, depending on the protocols and services being used. For example, you can configure the BIG-IP system to compress HTTP response data, or you can configure the system to authenticate SSL client certificates before passing requests on to a target server.

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, an SSL 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, SSL, and FTP, other profiles pertain to traffic behaviors applicable to multiple protocols. Examples of these are connection persistence profiles and authentication profiles. Table 5.1 lists the available profile types, with descriptions.

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Defines the behavior of Hypertext Transfer Protocol (HTTP) traffic.</td>
</tr>
<tr>
<td>FTP</td>
<td>Defines the behavior of File Transfer Protocol (FTP) traffic.</td>
</tr>
<tr>
<td>RTSP</td>
<td>Defines the behavior of Realtime Streaming Protocol (RTSP) traffic.</td>
</tr>
<tr>
<td>SIP</td>
<td>Defines the behavior of Session Initiation Protocol (SIP) traffic. The BIG-IP system routes SIP traffic based on this SIP profile, which persists on Call-ID; however, you can specify that SIP traffic persist on a value other than Call-ID by creating a SIP persistence profile, as shown following, under Persistence profiles.</td>
</tr>
</tbody>
</table>

Table 5.1 Available profile types on the BIG-IP system
### Profile Type | Description
--- | ---
**iSession** | Creates an optimization tunnel between two BIG-IP systems that are separated by a wide area network.

#### Persistence profiles

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookie</td>
<td>Implements session persistence using HTTP cookies.</td>
</tr>
<tr>
<td>Destination Address Affinity</td>
<td>Implements session persistence based on the destination IP address specified in the header of a client request. Also known as sticky persistence.</td>
</tr>
<tr>
<td>Hash</td>
<td>Implements session persistence in a way similar to universal persistence, except that the BIG-IP system uses a hash for finding a persistence entry.</td>
</tr>
<tr>
<td>Microsoft® Remote Desktop</td>
<td>Implements session persistence for Microsoft® Remote Desktop Protocol sessions.</td>
</tr>
<tr>
<td>SIP</td>
<td>Implements SIP message handling. Also, implements SIP persistence based on a specified SIP header field. Note that in order to use a SIP persistence profile, you must also create a SIP profile, as shown previously, under Services profiles.</td>
</tr>
<tr>
<td>Source Address Affinity</td>
<td>Implements session persistence based on the source IP address specified in the header of a client request. Also known as simple persistence.</td>
</tr>
<tr>
<td>SSL</td>
<td>Implements session persistence for non-terminated SSL sessions, using the session ID.</td>
</tr>
<tr>
<td>Universal</td>
<td>Implements session persistence using the BIG-IP system's Universal Inspection Engine (UIE).</td>
</tr>
</tbody>
</table>

#### Protocol profiles

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast L4</td>
<td>Defines the behavior of Layer 4 IP traffic.</td>
</tr>
<tr>
<td>Fast HTTP</td>
<td>Improves the speed at which a virtual server processes traffic.</td>
</tr>
<tr>
<td>HTTP Class</td>
<td>Forwards traffic to a destination based on examining traffic headers or content, using criteria that you specify.</td>
</tr>
<tr>
<td>TCP</td>
<td>Defines the behavior of TCP traffic.</td>
</tr>
<tr>
<td>UDP</td>
<td>Defines the behavior of UDP traffic.</td>
</tr>
<tr>
<td>SCTP</td>
<td>Defines the behavior of Streaming Control Transmission Protocol (SCTP) traffic.</td>
</tr>
</tbody>
</table>

#### SSL profiles

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Defines the behavior of client-side SSL traffic. See also Persistence Profiles.</td>
</tr>
<tr>
<td>Server</td>
<td>Defines the behavior of server-side SSL traffic. See also Persistence Profiles.</td>
</tr>
</tbody>
</table>

#### Authentication profiles

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>Allows the BIG-IP system to authenticate traffic based on authentication data stored on a remote Lightweight Directory Access Protocol (LDAP) server.</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Allows the BIG-IP system to authenticate traffic based on authentication data stored on a remote RADIUS server.</td>
</tr>
<tr>
<td>TACACS+</td>
<td>Allows the BIG-IP system to authenticate traffic based on authentication data stored on a remote TACACS+ server.</td>
</tr>
</tbody>
</table>

*Table 5.1 Available profile types on the BIG-IP system*
Understanding Profiles

Default profiles

The BIG-IP system includes one or more default profiles for each profile type listed in Table 5.1. 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.

### Table 5.1 Available profile types on the BIG-IP system

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL Client Certificate LDAP</td>
<td>Allows the BIG-IP system to control a client’s access to server resources based on data stored on a remote LDAP server. Client authorization credentials are based on SSL certificates, as well as defined user groups and roles.</td>
</tr>
<tr>
<td>SSL OCSP</td>
<td>Allows the BIG-IP system to check on the revocation status of a client certificate using data stored on a remote Online Certificate Status Protocol (OCSP) server. Client credentials are based on SSL certificates.</td>
</tr>
<tr>
<td>OneConnect</td>
<td>Enables client requests to reuse server-side connections. The ability for the BIG-IP system to reuse server-side connections is known as Connection Pooling™.</td>
</tr>
<tr>
<td>Statistics</td>
<td>Provides user-defined statistical counters.</td>
</tr>
<tr>
<td>NTLM</td>
<td>Uses an encrypted challenge/response protocol to authenticate a user without sending the user's password over the network.</td>
</tr>
<tr>
<td>Stream</td>
<td>Searches for and replaces strings within a data stream, such as a TCP connection.</td>
</tr>
</tbody>
</table>

Configuration Guide for BIG-IP® Local Traffic Management 5 - 3
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, and to manage client application authentication. 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.

◆ Note

You can manage only those profiles that you have permission to manage, based on your user role and partition access assignment.

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 6, Managing Application Layer Traffic
- Chapter 7, Enabling Session Persistence
- Chapter 8, Managing Protocol Profiles
- Chapter 9, Managing SSL Traffic
- Chapter 10, Authenticating Application Traffic
- Chapter 11, Using Additional Profiles

For background information on default and custom profiles, see Introducing profiles, on page 5-1.

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 5-12.

For information on creating or modifying a virtual server, see Chapter 2, 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.)

◆ Note

All default profiles reside in partition Common. For more information, see the TMOSTM Management Guide for BIG-IP® Systems.

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 5-12.

To modify a default profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The HTTP Profiles screen opens.
2. Select the default profile that you want to modify:
   - If you are modifying the http profile, click the name http.
     This displays the properties and settings of the default http profile.
   - 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.
     This displays the properties and settings of that default profile.
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 5-4.

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.

**Important**

When you create a custom profile, the BIG-IP system places the profile into your current administrative partition. For information on partitions, see the TMOS™ Management Guide for BIG-IP® Systems.

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 5-12.

**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 of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens and, by default, displays a list of any existing HTTP profiles.
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**.
   This displays the screen to create a new profile.
   
   **Note:** If the **Create** button is unavailable, this indicates that your user role does not grant you permission to create a profile.

4. In the **Name** box, type a unique name for your profile.

5. For the **Parent Profile** setting, 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 of the navigation pane, expand Local Traffic, and click Profiles.
   The HTTP Profiles screen opens.

2. From the menu for the type of profile you want to modify (Services, Persistence, Protocols, SSL, or Authentication), choose a profile type.
   This displays a list of existing profiles of that type.

3. In the Name column, click the name of the profile you want to modify.
   This displays the settings and values for that profile.

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.

◆ Note

You can manage only those profiles that you have permission to manage, based on your user role and partition access assignment.

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
- Partition in which the profile resides
- Type of profile (persistence and authentication 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 appears 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 of the navigation pane, expand Local Traffic, and click Profiles.
   The HTTP Profiles screen opens.
2. If you want to list profiles other than HTTP profiles, complete steps 3 and 4.
3. On the menu bar, click the category of profile you want to view. For example, if you want to view a list of TCP profiles, click Protocol.
4. From the menu, choose a profile type.
   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.
   A confirmation screen appears.
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 of the navigation pane, expand Local Traffic, and click Virtual Servers.
   The Virtual Servers screen opens.
2. In the Name column, click a virtual server name.
   This displays the properties and settings for that virtual server.
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.
Understanding Profiles

For example, a client application might use the TCP, SSL, 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, Client SSL, 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. Note that the BIG-IP system has specific requirements regarding the combinations of profile types allowed for a given virtual server.

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, SSL, and HTTP, and you have assigned SSL and HTTP profiles only, 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, Fast HTTP, or TCP profile type. Thus, if you have not associated a profile with the virtual server, the BIG-IP system adds a UDP, FastL4, Fast HTTP, 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.

Finding more information

For information on configuring specific types of profiles, see the following chapters:

- Chapter 6, Managing Application Layer Traffic
- Chapter 7, Enabling Session Persistence
- Chapter 8, Managing Protocol Profiles
- Chapter 9, Managing SSL Traffic
- Chapter 10, Authenticating Application Traffic
- Chapter 11, Using Additional Profiles
Managing Application Layer Traffic

- Introducing application layer traffic management
- Configuring HTTP standard profile settings
- Configuring HTTP compression
- Working with data compression strategies
- Configuring the RAM Cache feature
- Optimizing data compression and RAM caching
- Configuring FTP profile settings
- Configuring SIP profile settings
- Configuring RTSP profile settings
- Configuring iSession profile settings
Introducing application layer traffic management

The BIG-IP® local traffic management system offers several features that you can use to intelligently control your application layer 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 various configuration profiles. A profile is a group of settings, with values, that correspond 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.

To manage application layer traffic, you can use any of these profile types:

- HTTP (Hypertext Transfer Protocol)
- FTP (File Transfer Protocol)
- SIP (Session Initiation Protocol)
- RTSP (Real Time Streaming Protocol)
- iSession

**Note**

An additional profile type, SMTP, is available when BIG-IP® Protocol Security Module is licensed on the system. For information on configuring an SMTP profile, see the *Configuration Guide for BIG-IP® Protocol Security Module*.

In addition to these application layer 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 header or content data. For more information, see the remainder of this guide.

**Note**

There is another type of HTTP profile called a Fast HTTP profile. The use of a Fast HTTP profile is incompatible with the use of an HTTP profile. For more information on the Fast HTTP profile, see Chapter 8, *Managing Protocol Profiles*. 
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 5, Understanding Profiles.

For the profile settings that appear in the 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 HTTP compression, on page 6-12, and Configuring the RAM Cache feature, on page 6-30.

The BIG-IP system includes four specific HTTP profiles that you can implement:
- http
- http-acceleration
- http-wan-optimized-compression
- http-lan-optimized-caching
- http-wan-optimized-compression-caching

You can implement any one of these profiles as is, or you can create custom profiles by changing the value of the settings to suit your needs.

- If you are implementing a standard HTTP profile, see Understanding HTTP profile settings, following.
- If you are implementing the HTTP compression feature, see Configuring HTTP compression, on page 6-12.
- If you are implementing the RAM Cache feature, see Configuring the RAM Cache feature, on page 6-30.
- If you are implementing one of the HTTP optimization profiles, see Optimizing data compression and RAM caching, on page 6-36.

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 6.1, on page 6-3 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.
### Table 6.1 Settings of an HTTP profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the user-supplied name of the profile. You must specify a name for your profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which your custom profile is derived.</td>
<td>http</td>
</tr>
<tr>
<td>Basic Auth Realm</td>
<td>Specifies an authentication realm for client authentication.</td>
<td>No default value</td>
</tr>
<tr>
<td>Fallback Host</td>
<td>Specifies the fallback host to send as an HTTP 302 response when either all nodes are down, or the selected node is down but not yet detected as such by the associated monitors.</td>
<td>No default value</td>
</tr>
<tr>
<td>Fallback on Error Codes</td>
<td>Specifies the HTTP error codes that cause the BIG-IP system to redirect an HTTP response to the host specified with the <strong>Fallback Host</strong> setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Request Headers Insert</td>
<td>Specifies the header string that you want to insert into an HTTP request.</td>
<td>No default value</td>
</tr>
<tr>
<td>Request Headers Erase</td>
<td>Specifies the header string that you want to erase from an HTTP request.</td>
<td>No default value</td>
</tr>
<tr>
<td>Response Headers Allowed</td>
<td>Specifies headers that the BIG-IP system allows in an HTTP response.</td>
<td>No default value</td>
</tr>
<tr>
<td>Response Chunking</td>
<td>Specifies how to handle chunking for HTTP responses. Possible values are <strong>Unchunk</strong>, <strong>Rechunk</strong>, <strong>Selective</strong>, and <strong>Preserve</strong>.</td>
<td>Selective</td>
</tr>
<tr>
<td>OneConnect Transformations</td>
<td>Performs HTTP header transformations for the purpose of keeping connections open. This feature requires configuration of a One Connect™ profile.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Redirect Rewrite</td>
<td>Allows you to modify HTTP redirections. Possible values are <strong>Matching</strong>, <strong>All</strong>, <strong>Nodes</strong>, or <strong>None</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>Encrypt Cookies</td>
<td>Encrypts specified cookies that the BIG-IP system sends to a client system.</td>
<td>No default value</td>
</tr>
<tr>
<td>Confirm Cookie Encryption Passphrase</td>
<td>Confirms the passphrase that you specified in the <strong>Cookie Encryption Passphrase</strong> box.</td>
<td>No default value</td>
</tr>
<tr>
<td>Maximum Header Size</td>
<td>Specifies the maximum size in bytes that the BIG-IP system allows for HTTP headers.</td>
<td>32768</td>
</tr>
<tr>
<td>Pipelining</td>
<td>Enables or disables HTTP pipelining.</td>
<td>Enabled</td>
</tr>
</tbody>
</table>
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.

### Specifying a realm for basic authentication

The value of the Basic Auth Realm setting is a string that you provide. The BIG-IP system sends this string to a client as part of client authentication.

To configure this setting, locate the Basic Auth Realm setting and type a value.

---

**Table 6.1 Settings of an HTTP profile**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert XForwarded-For</td>
<td>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.</td>
<td>Disabled</td>
</tr>
<tr>
<td>LWS Maximum Columns</td>
<td>Specifies the maximum width allowed for an HTTP header that is inserted into an HTTP request.</td>
<td>80</td>
</tr>
<tr>
<td>LWS Separator</td>
<td>Specifies the separator that the BIG-IP system should use between HTTP headers when a header exceeds the maximum width allowed.</td>
<td>\n</td>
</tr>
<tr>
<td>Maximum Requests</td>
<td>Specifies the maximum number of HTTP requests that the system allows for a single Keep-Alive connection.</td>
<td>0</td>
</tr>
<tr>
<td>Protocol Security</td>
<td>Specifies, when checked (enabled), that the system inspects HTTP traffic for security vulnerabilities, by using a security profile in the Protocol Security module. This option is available only when you are licensed for the module.</td>
<td>Unchecked</td>
</tr>
</tbody>
</table>
Specifying a fallback host

Another feature that you can configure within an HTTP profile is HTTP redirection. **HTTP redirection** allows you to redirect HTTP traffic to another protocol identifier, host name, port number, or URI path.

Redirection to a fallback host occurs if all members of the targeted pool are unavailable, or if a selected pool member is unavailable. (The term **unavailable** refers to a member being disabled, marked as **down**, or having exceeded its connection limit.) When one or more pool members are unavailable, the BIG-IP system can redirect the HTTP request to the fallback host, with the HTTP reply **Status Code 302 Found**.

Although HTTP redirection often occurs when the system generates an **LB_FAILED** iRule event, redirection can also occur without the occurrence of this event, such as when:

- The selected node sends an **RST** after a **TCP 3WHS** has completed, but before the node has sent at least a full response header.
- The BIG-IP system finds the selected node to be unreachable while receiving the body portion of a request or a pipelined request.

When configuring the BIG-IP system to redirect HTTP traffic to a fallback host, you can specify an IP address or a fully-qualified domain name (FQDN). The value that you specify becomes the value of the **Location** header that the servers sends in the response. For example, you can specify a redirection as **http://redirector.siterequest.com**.

Specifying fallback error codes

In addition to redirecting traffic when a target server becomes unavailable, you can also specify the HTTP error codes from server responses that should trigger a redirection to the fallback host. Typical error codes to specify in the **Fallback Error Codes** setting are **500**, **501**, and **502**.

When you type the error codes in the **Fallback Error Codes** box, you separate the codes with a space, such as **500 501 502**. You can also specify a range of error codes, as in this example: **505-515**.

Inserting headers into HTTP requests

An optional setting in an HTTP profile is HTTP header insertion. The HTTP header being inserted can include a client IP address. Including a client IP address in an HTTP header is useful when a connection goes through a secure network address translation (SNAT) and you need to preserve the original client IP address.

The format of the header insertion that you specify is generally a quoted string. Alternatively, however, you can insert a Tools Command Language (Tcl) expression into a header that dynamically resolves to the desired value.
When you assign the configured HTTP profile to a virtual server, the BIG-IP system then inserts the header specified in the profile into any HTTP request that the BIG-IP system sends to a pool or pool member.

◆ Note

In addition to inserting a string such as a client IP address into an HTTP request, you can configure the BIG-IP system to insert SSL-related headers into HTTP requests. Examples are: client certificates, cipher specifications, and client session IDs. To insert these types of headers, you must create an iRule. For more information on using iRule commands to perform header insertion, see Chapter 17, Writing iRules, and the F5 Networks DevCentral web site, http://devcentral.f5.com.

To insert a header into an HTTP request, locate the Request Headers Insert setting and type a value.

Erasing content from HTTP headers

Another optional setting is the Header Erase setting. Using this setting, you can configure a profile to erase the contents of a header from an HTTP request that is being sent from a client to a server. With this feature, you can erase header content from HTTP requests before forwarding the requests over the network. Such headers might contain sensitive information, such as user IDs or telephone numbers, that must be erased before the information is forwarded.

When you use this setting, the BIG-IP system erases the contents of the specified header and replaces that content with blank spaces. The header itself is retained.

◆ Note

This feature does not apply to HTTP responses being sent from a server to a client.

The client header with the contents to be erased must be specified as a quoted string. To erase a header from an HTTP request, locate the Request Headers Erase setting and type a value.

Allowing headers in an HTTP response

Using the Configuration utility, you can specify any headers within an HTTP response that you want the BIG-IP system to allow. To do this, type a value in the Response Headers Allowed box. If you are specifying more than one header, separate the headers with a blank space. For example, if you type the string Content-Type Set-Cookie Location, the BIG-IP system then allows the headers Content-Type, Set-Cookie, and Location.
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 6.2 describes each of these values and the action that the BIG-IP system takes, depending on whether an original response is chunked or unchunked.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Original response is chunked</th>
<th>Original response is unchunked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unchunk</td>
<td>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 <strong>Connection: Close</strong> header.</td>
<td>The BIG-IP system processes the HTTP content and passes the response on untouched.</td>
</tr>
<tr>
<td>Rechunk</td>
<td>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.</td>
<td>The BIG-IP system adds transfer encoding and chunking headers on egress.</td>
</tr>
<tr>
<td>Selective</td>
<td>Same as Rechunk.</td>
<td>The BIG-IP system processes the HTTP content and then passes the response on untouched.</td>
</tr>
<tr>
<td>Preserve</td>
<td>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.</td>
<td>The BIG-IP system processes the HTTP content and then passes the response on untouched.</td>
</tr>
</tbody>
</table>

*Table 6.2 Chunking behavior of the BIG-IP system*

Enabling or disabling OneConnect transformations

This setting enables or disables part of the OneConnect™ feature and applies to HTTP/1.0 connections only. When this setting is enabled and a OneConnect profile is assigned to the virtual server, the setting performs **Connection** header transformations, for the purpose of keeping a client connection open. More specifically:

1. A client sends an HTTP/1.0 request.
2. The server sends a response, which initially includes a **Connection: Close** header.
3. The BIG-IP system transforms the Connection: Close header to Connection: Keep-Alive.

4. Through use of the OneConnect profile, the server-side connection detaches, goes into the pool of available server-side connections used for servicing other requests, and eventually closes. This process is hidden from the client.

5. The client-side connection remains open, operating under the assumption that the server-side connection is still open and therefore able to accept additional requests from that client.

The default value for this setting is Enabled. To enable or disable OneConnect transformations, locate and enable the OneConnect Transformations setting.

◆ Important

For this setting to take effect, you must also configure a OneConnect™ profile, which enables connection pooling. For more information on connection pooling and configuring a OneConnect profile, see Chapter 5, Understanding Profiles.

For general information on the OneConnect™ feature, see Chapter 1, Introduction to Local Traffic Management.

Rewriting an HTTP redirection

Sometimes, a client request is redirected from the HTTPS protocol to the HTTP protocol, which is a non-secure channel. If you want to ensure that the request remains on a secure channel, you can cause that redirection to be rewritten so that it is redirected back to the HTTPS protocol.

Note that the rewriting of any redirection takes place only in the HTTP Location header of the redirection response, and not in any content of the redirection.

To enable the BIG-IP system to rewrite HTTP redirections, you simply specify, through the Configuration utility, the way that you want the system to handle URIs during the rewrite. Once enabled, this feature rewrites the protocol name port number

Possible values for this setting are Matching, All, Nodes, or None.

Selecting URIs to rewrite

When configuring the BIG-IP system to rewrite HTTP redirections, you specify whether the system should rewrite only those URIs matching the URI originally requested by the client (minus an optional trailing slash), or whether the system should rewrite all URIs. In the latter case, the system always rewrites redirected-to URIs, and rewrites those URIs as if they matched the originally-requested URIs.

If the URI contains a node IP address instead of a host name, you can configure the BIG-IP system to change that IP address to the virtual server address.
Table 6.3 shows examples of how redirections of client requests are transformed when the BIG-IP system is listening on port 443, and the **Rewrite Redirections** setting is enabled.

<table>
<thead>
<tr>
<th>Original Redirection</th>
<th>Rewrite of Redirection</th>
</tr>
</thead>
</table>

*Table 6.3  Examples of rewriting HTTP redirections with the system listening on port 443*

Table 6.4 shows examples of how redirections of client requests are transformed when the system is listening on port 4443, and the rewrite feature is enabled.

<table>
<thead>
<tr>
<th>Original Redirection</th>
<th>Rewrite of Redirection</th>
</tr>
</thead>
</table>

*Table 6.4  Examples of rewriting HTTP redirections with the system listening on port 4443*

**Rewriting the protocol name**

When configured to rewrite HTTP redirections, the BIG-IP system rewrites the HTTP protocol name to HTTPS. For example, a client might send a request to `https://www.sample.com/bar` and be initially redirected to `http://www.sample.com/bar/`, which is a non-secure channel. If you want the client request to remain on a secure channel, you can configure the BIG-IP system to rewrite the redirected URI to go to `https://www.sample.com/bar/` instead. (Note the addition of the trailing slash.)

**Encrypting and decrypting cookies**

You can use the Configuration utility to encrypt one or more cookies that the BIG-IP system sends to a client system. When the client sends the encrypted cookie back to the BIG-IP system, the system decrypts the cookie.

To encrypt a cookie, use the **Encrypt Cookie** setting to specify the name of the cookie that you want to the BIG-IP system to encrypt. If you want to specify more than one cookie, simply separate the cookie names with a space.

After specifying one or more cookie names, use the **Cookie Encryption Passphrase** and the **Confirm Cookie Encryption Passphrase** boxes to type a passphrase for the cookie.
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 initiate a request until the previous request has received a response. HTTP/1.1 **pipelining** allows clients to initiate multiple requests even when prior requests have not received a response. Note, however, that each initiated request is still processed sequentially; that is, a request in the queue is not processed until the previous request has received a response.

By enabling support for pipelining on the BIG-IP system, you remove the need to enable pipelining on the destination server itself.

To enable or disable 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 HTTP compression

In a typical client-server scenario, browsers and servers can be configured to compress and uncompress HTTP content. HTTP compression reduces the amount of data to be transmitted, thereby significantly reducing bandwidth usage. The following two sections describe the benefits of configuring the BIG-IP system to perform HTTP compression tasks.

Compression in a typical client-server scenario

When HTTP compression is enabled in a client-server environment, a browser inserts an `Accept-Encoding` header into a client request, specifying the compression methods that the browser understands. Consequently, the server reads the header and compresses the response body with one of those compression methods. The server then inserts a `Content-Encoding` header in the response, to communicate to the browser the compression method that the server used. Upon receiving the compressed response, the browser reads the `Content-Encoding` header and uncompresses the data accordingly.

Enabling HTTP compression without a BIG-IP system typically requires installation and configuration of compression software on the destination server.

Compression using the BIG-IP system

An optional feature is the BIG-IP system’s ability to off-load HTTP compression tasks from the target server. All of the tasks needed to configure HTTP compression on the BIG-IP system, as well as the compression software itself, are centralized on the BIG-IP system.

When configuring the system to compress data, you can also:

- Configure the system to include or exclude certain types of data
- Specify the levels of compression quality and speed that you want

The primary way to enable the HTTP compression option is by setting the Compression setting of an HTTP profile to Enabled. This causes the BIG-IP system to compress HTTP content for any responses in which the values that you specify in the URI List or Content List settings of an HTTP profile match the values of the Request-URI or Content-Type response headers.

◆ Note

The string that you specify in the URI List or the Content List setting can be either a pattern string or a regular expression. Note that list types are case-sensitive for pattern strings. For example, the system treats the pattern string `www.f5.com` differently from the pattern string `www.F5.com`. You can override this case-sensitivity by using the Linux regexp command.
Important

When you enable compression by configuring an HTTP profile, the system uses a default compression strategy named Speed. The Speed compression strategy ensures that the BIG-IP system uses the hardware compression provider whenever possible, to maximize performance. You can change the compression strategy by using a command line interface. For a detailed description of the available compression strategies, see Working with data compression strategies, on page 6-23.

If you want to enable HTTP compression for specific connections, you can write an iRule that specifies the HTTP:compress enable command. For more information, see Chapter 17, Writing iRules, and the F5 Networks DevCentral web site, http://devcentral.f5.com.

When HTTP compression is enabled on the BIG-IP system, the BIG-IP system performs a series of steps:

1. First, the BIG-IP system reads the Accept-Encoding header of a client request, looks for specification of either the deflate or gzip compression method, and notes whether either method is marked as being preferred.

2. If the Keep Accept Encoding setting in the HTTP profile is set to Disabled, the BIG-IP system then removes the Accept-Encoding header from the request and passes the request on to the server. Removing the Accept-Encoding header prevents the server from performing the HTTP compression and from inserting the Content-Encoding header into its response.

3. Upon receiving the server response, the BIG-IP system inserts the Content-Encoding header, specifying the compression method that it has chosen to use. The BIG-IP system chooses a compression method by looking for the specification of either the gzip or deflate compression method in the Accept-Encoding header of the client request. If the client request does not specify a compression method, the BIG-IP system can use either the gzip or deflate method to compress the response data. In this case, the default method that the BIG-IP system uses is gzip.

4. Finally, the BIG-IP system then compresses the response and sends it to the client. The client then reads the Content-Encoding header in the response, determines the compression method used, and uncompresses the data accordingly.

Using the BIG-IP system HTTP compression feature, you can include or exclude certain types of URIs or files that you specify. Exclusion is useful because some URI or file types might already be compressed. Using CPU resources to compress already-compressed data is not recommended because the cost of compressing the data usually outweighs the benefits. Examples of regular expressions that you might want to specify for exclusion are .pdf, .gif, or .html.
Alternatives to configuring data compression

Although creating a custom HTTP profile is the primary way to implement data compression on the BIG-IP system, there are other ways to implement data compression:

- You can implement the default `http-wan-optimized-compression` profile as is. This profile contains settings that are already configured to optimize data compression. This is a simpler configuration task, but it gives you less control over the type of data being compressed. For more information, see Using the `http-wan-optimized-compression` profile, on page 6-37.

- You can create a custom profile based on the `http-wan-optimized-compression` profile, and then modify the compression settings. For more information, see Using the `http-wan-optimized-compression` profile, on page 6-37.

- For a more advanced compression implementation, you can create a custom profile based on the default `http` profile, and then configure a different compression strategy using a command line interface. Different compression strategies affect the way that the system uses hardware and software compression providers to provide the most efficient compression possible. For more information, see Working with data compression strategies, on page 6-23.

Using an HTTP profile for data compression

As previously described, one way to implement data compression is to create a custom profile based on the default `http` profile. Table 6.5 shows the compression settings that you can specify within a custom HTTP profile that you create. Configuring these settings means either specifying a value where no default value exists, or changing a default value.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>Enables or disables the HTTP compression feature.</td>
<td>Disabled</td>
</tr>
<tr>
<td>URI Compression</td>
<td>Displays the settings for including or excluding certain Request-URI responses. Possible values are URI List or Not Configured.</td>
<td>Not Configured</td>
</tr>
<tr>
<td>URI List</td>
<td>If the URI Compression setting is set to Enabled, specifies the URI targeted for compression, as well as the types of responses to include for and exclude from URI compression. <strong>Note:</strong> When you use pattern strings, this list type is case-sensitive. For more information, see the previous section of this chapter.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

Table 6.5 Compression settings of an HTTP profile
Content Compression Displays the settings for including or excluding certain Content-Type responses. Possible values are Content List or Not Configured.

Content List If the Content Compression setting is set to Enabled, specifies the type of content targeted for compression, as well as the type of responses to include for or exclude from content compression. **Note:** When you use pattern strings, this list type is case-sensitive. For more information, see the previous section of this chapter.

Preferred Method Specifies the compression method that you want to use to compress the response. Possible values are gzip and deflate.

Minimum Content Length Specifies the minimum length in bytes of a server response that is acceptable for compressing that response. The length applies to content length only, not headers.

Compression Buffer Size Specifies the maximum number of compressed bytes that the BIG-IP system buffers before deciding whether or not to insert a Content-Length header into the response that specifies the compressed size.

gzip Compression Level Specifies the amount and rate of compression. 1 - Least Compression (Fastest)

gzip Memory Level Specifies the number of kilobytes of memory that the BIG-IP system uses for internal compression buffers when compressing a server response. 8

gzip Window Size Specifies the number of kilobytes in the window size that the BIG-IP system uses when compressing a server response. 16

Vary Header Enables or disables the insertion of a Vary header into cacheable server responses. Enabled

HTTP/1.0 Requests Enables or disables compression of responses to HTTP/1.0 client requests. Disabled

Keep Accept Encoding When enabled, allows the target server to perform the HTTP compression instead of the BIG-IP system. Disabled

Browser Workarounds Implements browser workarounds. Disabled (Cleared)

**Table 6.5** Compression settings of an HTTP profile
Before configuring an HTTP profile, it is helpful to have a description of the compression settings that you might want to change.

### Enabling or disabling the compression feature

The **Compression** setting causes the BIG-IP system to perform compression on server responses. Possible values are:

- **Enabled**
  - This value causes the BIG-IP system to compress, or refrain from compressing, HTTP server content for any responses matching the values that you specify in the **URI List** or **Content List** settings of the HTTP profile.

- **Disabled**
  - When the **Compression** setting is set to **Disabled**, the BIG-IP system does not perform HTTP compression.

- **Selective**
  - This setting causes the BIG-IP system to perform HTTP compression only when specified by an iRule containing the `HTTP::compress` command.

To enable HTTP compression, locate the **Compression** setting and select **Enabled**. If you are enabling compression through the use of the iRule command `HTTP::compress <enable>`, select **Selective**.

### Using URI compression

If you enable compression, you probably do not want the BIG-IP system to compress every kind of server response. Using the **URI Compression** setting, you can set its value to **URI List**, which instructs the BIG-IP system to include in compression, or exclude from compression, certain responses that are specified in the URIs of client requests.

---

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Saver</td>
<td>Specifies, when checked (enabled), that the system monitors the percent CPU usage and adjusts compression rates automatically when the CPU usage reaches either the <strong>CPU Saver High Threshold</strong> or the <strong>CPU Saver Low Threshold</strong>.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>CPU Saver High Threshold</td>
<td>Specifies the amount of CPU usage that causes the system to change the amount of content being compressed, and the amount of compression being applied.</td>
<td>90</td>
</tr>
<tr>
<td>CPU Saver Low Threshold</td>
<td>Specifies the amount of CPU usage that causes the system to revert back to user-defined compression values.</td>
<td>75</td>
</tr>
</tbody>
</table>

**Table 6.5 Compression settings of an HTTP profile**
More specifically, you can type regular expressions to specify the types of server responses that you want the BIG-IP system to include in, or exclude from, compression. For example, you can specify that you want the BIG-IP system to compress all .htm responses by typing the regular expression ".*htm". The BIG-IP system then compares that response type to the URI specified within each client request, and if the system finds a match, takes some action.

Any regular expression that you specify must be in Advanced Regular Expression (ARE) syntax.

To use the URI Compression feature, locate the URI Compression setting and select URI List. You can then use the Include List box or the Exclude List box to type your regular expressions. If you do not specify any list (either URI or content), the BIG-IP system compresses all responses. For more information on content lists, see Using content compression, on page 6-17.

Including URI-specified responses in HTTP compression

When the URI compression setting is enabled, and you type one or more values in the Include List, the BIG-IP system compresses only those responses that match the URI part of the client request line.

The values you specify in the Include List box are in the form of regular expressions. For example, if the Include List box contains the values ".*txt", ".*htm", and ".*html", and those expressions match URIs in client requests, then the BIG-IP system compresses only the responses for those URIs that match the specified regular expressions.

To apply this setting successfully, the BIG-IP system must find a match in at least one of the values specified in the Include List box. If the BIG-IP system finds no match, then no response is compressed.

Excluding URI-specified responses in HTTP compression

When the URI compression setting is enabled, and you type one or more values in the Exclude List, the BIG-IP system excludes from compression those responses that match the URI part of the client request line.

The values you specify in the Exclude List box are in the form of regular expressions. For example, if the Exclude List box contains the value ".*pdf", and that expression matches URIs in client requests, then the BIG-IP system excludes from compression any .pdf responses for those URIs.

To apply this setting successfully, the BIG-IP system must find a match in at least one of the values specified in the Exclude List box. If the BIG-IP system finds no match, then no responses are excluded from compression.

Using content compression

If you enable compression, you probably do not want the BIG-IP system to compress every kind of server response. Using the Content Compression setting, you can set its value to Content List, which instructs the BIG-IP
system to include in compression, or exclude from compression, certain responses that are specified in the Content-Type header of server responses.

More specifically, you can type regular expressions to specify the types of server responses that you want the BIG-IP system to include in, or exclude from, compression. For example, you can specify that you want the BIG-IP system to compress all .htm responses by typing the regular expression .*.htm. The BIG-IP system then compares that response type to the value of the Content-Type header specified within each server response, and if the system finds a match, takes some action.

Any regular expression that you specify must be in Advanced Regular Expression (ARE) syntax.

To use the Content Compression feature, locate the Content Compression setting and select Content List. You can then use the Include List box or the Exclude List box to type your regular expressions. If you do not specify any list (either URI or content), the BIG-IP system compresses all responses. For more information on content lists, see Using URI compression, on page 6-16.

Including Content-Type responses in HTTP compression

When compression is enabled and you specify one or more values in the Include List box, the BIG-IP system includes only those responses that match the value of the server’s Content-Type header. The value of this setting is a list of those header values. For example, if the Include List box contains the values application/pdf and image/**, then only responses of those content types are compressed.

To include all text types, you assign the value text/* to this setting.

Excluding Content-Type responses in HTTP compression

When compression is enabled and you specify one or more values in the Exclude List box, the BIG-IP system excludes those responses that match the value of the server’s Content-Type header. The value of this setting is a list of those header values. For example, if the Exclude List box contains the values application/pdf and image/**, then responses of those content types are excluded from compression.

To exclude all text types, you assign the value text/* to this setting.

Specifying a preferred compression method

Using the Preferred Method setting, you can specify the compression method that you want the BIG-IP system to use when compressing responses. The two possible values are gzip and deflate. The default value is gzip.
Specifying minimum content length for compression

When compression is enabled, the Minimum Content Length setting specifies the minimum length of a server response in uncompressed bytes that the BIG-IP system requires for compressing that response. The BIG-IP system finds the content length of a server response in the Content-Length header of the server response. Thus, if the content length specified in the response header is below the value assigned to the Minimum Content Length setting, the BIG-IP system does not compress the response. The length in bytes applies to content length only, not headers.

For example, using the default value of 1024, the BIG-IP system compresses only those responses with HTTP content containing at least 1024 bytes. Sometimes the Content-Length header does not indicate the content length of the response. In such cases, the BIG-IP system compresses the response, regardless of size.

To specify a minimum content length, locate the Minimum Content Length setting, and type a numeric value.

Specifying the compression buffer size

When compression is enabled, the Compression Buffer Size setting specifies the maximum number of compressed bytes that the BIG-IP system buffers before deciding whether or not to preserve a Keep-Alive connection and rewrite the Content-Length header.

For example, using the default value of 4096, the BIG-IP system buffers up to 4096 bytes of compressed data before deciding whether or not to preserve the connection and rewrite the Content-Length header.

The BIG-IP system’s decision to rewrite the Content-Length header depends on whether response chunking is enabled (using the Response Chunking profile setting). Table 6.6, shows the behavior of the BIG-IP system with respect to compression buffer size and response chunking.

<table>
<thead>
<tr>
<th>If the size of the compressed response is</th>
<th>And the compressed response is</th>
<th>Then the BIG-IP system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal to or greater than the maximum buffer size</td>
<td>Chunked</td>
<td>Keeps the connection open (if the Connection header is not set to Close).</td>
</tr>
<tr>
<td></td>
<td>Not chunked</td>
<td>Closes the connection by changing the value of the Connection header to Close.</td>
</tr>
<tr>
<td>Less than the maximum buffer size</td>
<td>Chunked</td>
<td>Does not insert a Content-Length header with a compressed response size.</td>
</tr>
<tr>
<td></td>
<td>Not chunked</td>
<td>Inserts a Content-Length header with a compressed response size.</td>
</tr>
</tbody>
</table>

Table 6.6  BIG-IP system behavior based on maximum buffer size
For more information, see *Specifying minimum content length for compression*, preceding.

To specify a compression buffer size, locate the **Compression Buffer Size** setting and type a numeric value.

### Specifying a compression level

Using the **gzip Compression Level** setting, you can specify the extent to which data is compressed, as well as the compression rate. Possible values are 1 - Least Compression (Fastest), 9 - Most Compression (Slowest), and Other. The default compression level is 1 - Least Compression (Fastest).

You can specify any whole number ranging from 1 through 9, with these results:

- Specifying a lower number causes data to be less compressed but at a higher performance rate. Thus, a value of 1 causes the least compression but the fastest performance.
- Specifying a higher number causes data to be more compressed but at a slower performance rate. Thus, a value of 9 (the highest possible value) causes the most compression, but the slowest performance.

**WARNING**

Selecting any value other than 1 - Least Compression (Fastest) can degrade system performance.

**Tip**

You can change the way that the BIG-IP system uses gzip levels to compress data. You do this by configuring the compression strategy. The compression strategy determines the particular compression provider (hardware or software) that the system uses for HTTP responses. The available strategies are: Speed (the default strategy), Size, Ratio, and Adaptive. For information on how to configure the compression strategy, see *Working with data compression strategies*, on page 6-23.

### Specifying a memory level for gzip compression

The **gzip Memory Level** setting specifies a value representing the amount of kilobytes of memory that the BIG-IP system uses to compress data when using the gzip or deflate compression method. The value of the **gzip Memory Level** setting must be a power-of-2 integer, in bytes, ranging from 1 to 256.

Generally, a higher value causes the BIG-IP system to use more memory, but results in a faster and higher compression ratio. Conversely, a lower value causes the BIG-IP system to use less memory, but results in a slower and lower compression ratio. The default value is 8.

To specify a memory level, locate the **gzip Memory Level** setting and select a numeric value.
Specifying window size for gzip compression

The gzip Window Size setting specifies a value representing the number of kilobytes in window size that the BIG-IP system uses when compressing a server response using the gzip or deflate compression method. The value of the gzip Window Size setting must be a power-of-2 integer, in bytes, ranging from 1 to 128.

Generally, a higher value causes the BIG-IP system to use more memory, but results in a higher compression ratio. Conversely, a lower value causes the BIG-IP system to use less memory, but results in a lower compression ratio. The default value is 16.

To specify a window size, locate the gzip Window Size setting and select a numeric value.

Enabling or disabling the Vary header

When compression is enabled, the Vary Header setting inserts the Vary: Accept-Encoding header into a compressed server response. If the Vary header already exists in the response, the BIG-IP system appends the value Accept-Encoding to that header.

The reason for inserting the Vary: Accept-Encoding header into a server response is to follow a recommendation by RFC2616, which states that the Vary header should be inserted into any cacheable response that is subject to server-driven negotiation. Server responses that are subject to HTTP compression fall into this category.

If the Vary Header setting is disabled, the BIG-IP system does not insert the Vary header into a server response.

To disable the Vary header, locate the Vary Header setting and clear the Enabled box.

Allowing compression for HTTP/1.0 requests

The HTTP/1.0 Requests setting is included for backward compatibility, allowing HTTP compression for responses to HTTP/1.0 client requests. The default value for this setting is Disabled.

If this setting is set to Enabled, the BIG-IP system only compresses responses in either of the following cases:

• When the server responds with a Connection: close header
• When the response content is no greater than the value of the Compression Buffer Size setting

To enable compression for HTTP/1.0 requests, locate the HTTP/1.0 Requests setting and check the box.
Keeping the Accept-Encoding header

Normally, when you enable HTTP compression, the BIG-IP system strips out the **Accept-Encoding** header from the HTTP request. This causes the BIG-IP system to perform the HTTP compression instead of the target server.

By default, the **Keep Accept Encoding** setting is disabled. If you want to allow the target server instead of the BIG-IP system to perform the HTTP compression, simply enable this setting.

Implementing browser workarounds

When you enable the **Browser Workarounds** setting, the system uses built-in workarounds for several common browser issues that occur when compressing content. The default setting is disabled (cleared). More specifically, enabling this setting prevents the system from compressing server responses when any of these conditions exist:

- The client browser is Netscape® version 4.0x.
- The client browser is Netscape version 4.x (that is, versions 4.10 and higher), and the **Content-Type** header of the server response is not set to **text/html** or **text/plain**.
- The client browser is Microsoft® Internet Explorer® (any version), the **Content-Type** header of the server response is set to either **text/css** or **application/x-javascript**, and the client connection uses SSL.
- The client browser is Microsoft® Internet Explorer® (any version), the **Content-Type** header of the server response is set to either **text/css** or **application/x-javascript**, and the **Cache-Control** header of the server response is set to **no-cache**.

CPU Saver

When you enable the **CPU Saver** setting, the system monitors the percent CPU usage and adjusts compression rates automatically when the CPU usage reaches the percentage defined in either the **CPU Saver High Threshold** or the **CPU Saver Low Threshold**. The default setting is enabled (checked).

**CPU Saver High Threshold**

The **CPU Saver High Threshold** setting specifies the percent of CPU usage at which the system starts automatically decreasing the amount of content being compressed, as well as the amount of compression which the system is applying. The default setting is 90 percent.

**CPU Saver Low Threshold**

The **CPU Saver Low Threshold** specifies the percent CPU usage at which the system resumes content compression at the user-defined rates. The default setting is 75 percent.
Working with data compression strategies

If you need more control over the way that the BIG-IP system compresses data than what the standard HTTP profile configuration provides, you can enable a compression strategy other than the default strategy. The default compression strategy is Speed.

This section describes the compression providers available on the BIG-IP system, the four available compression strategies, and detailed information on using the Adaptive compression strategy in particular.

◆ Note

A BIG-IP command line interface is the only tool available for setting the compression strategy. If you do not explicitly set the strategy, or you are using the Configuration utility rather than the command line interface to configure compression, the BIG-IP system uses the Speed strategy as the default strategy.

Understanding compression providers

The BIG-IP system utilizes compression providers to compress HTTP server responses. The compression providers are either hardware cards or software programs that you can install on multiprocessor BIG-IP systems to perform HTTP data compression. The software and hardware compression providers that the BIG-IP system can use to compress data are:

◆ A hardware compression card
   A hardware compression card increases the speed at which data is compressed. A hardware compression card is optional on the 6400, 6800, and 8400 BIG-IP platforms. On the 8800 platform, a hardware compression card is always included.

◆ zlib
   The zlib tool is a software compression provider included in the BIG-IP system software. The BIG-IP system uses zlib only when the hardware compression card is too busy to compress additional data.

It important to understand that hardware compression providers cannot match the highest quality compression level that software compression providers perform. Conversely, software compression providers require more system resources than hardware compression providers to deliver the highest quality compression.

You can view the hardware compression providers that are present on your system using the bigpipe utility.
Understanding compression strategy selection

When using the command line interface to configure compression for a BIG-IP system, you can choose from four compression strategies: Speed, Size, Ratio, and Adaptive. The BIG-IP system uses the compression strategy that you select to determine which compression provider to use for a given HTTP response. Once an HTTP response is assigned to a compression provider, the response remains associated with that compression provider until the response is completed.

Table 6.7 describes the four compression strategies.

<table>
<thead>
<tr>
<th>Compression Strategies</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>This is the default compression strategy. The system uses the hardware compression provider to the fullest extent possible. Only when the hardware is busy does the system use a software compression provider to compress HTTP server responses. The Speed strategy is best used for bulk compression and for limiting CPU overhead.</td>
</tr>
<tr>
<td>Size</td>
<td>The system performs as much compression in the software as possible using a ratio of TMM and Offload. Only when the software is busy does the system use the hardware compression provider to compress HTTP server responses. The Size strategy gives the best ratio at the expense of CPU overhead.</td>
</tr>
<tr>
<td>Ratio</td>
<td>The system uses a weighted Round Robin approach to decide which compression provider to use to compress data. The Ratio strategy limits CPU overhead while giving good compression ratios.</td>
</tr>
<tr>
<td>Adaptive</td>
<td>The system first uses a software compression provider to compress HTTP server responses. The system switches to the hardware compression providers based on both the gzip compression level that you set in the HTTP profile and the hardware compression provider that the system contains. As load on the system increases, the system responds by reducing the desired gzip compression level (specified in the HTTP profile). The system utilizes the hardware compression provider only when the provider can deliver the specified or systematically-reduced gzip compression level. The Adaptive strategy gives you the most control over how the BIG-IP system handles compression.</td>
</tr>
</tbody>
</table>

Table 6.7 Compression strategies described

Understanding the adaptive compression strategy

The adaptive compression strategy specifies that the BIG-IP system can use both software and hardware compression providers in the most efficient way possible to provide the best quality of compression, while still ensuring availability of resources for load balancing. When adaptive compression is enabled, the system selects from the available compression providers to compress HTTP server responses based on a desired level of quality (gzip level) that you specify. Adaptive compression provides the most benefit when you have a BIG-IP 6400, 6800, or 8400 system that contains a hardware compression card.
To use adaptive compression, you perform the following tasks:

- Enable adaptive compression on the system.
- Create an HTTP profile and set the gzip compression level.

For more information on configuring adaptive compression, see *Configuring adaptive compression*, on page 6-28.

Setting the adaptive compression strategy causes the following effect on the system:

- When the load on the system decreases, the adaptive compression strategy allows the system to incrementally increase the quality of the compression of server responses.
- When the load on the system increases, the adaptive compression strategy allows the system to incrementally decrease the quality of the compression of server responses. This frees the system resources to handle the load balancing of the increased traffic.
- When traffic reaches a peak volume, and based on the gzip compression level that you set in the HTTP profile, the system begins to compress data using its hardware compression provider.

### Understanding platform differences

When you configure the adaptive compression strategy, the particular compression providers that the BIG-IP system uses depends not only on the gzip level you have configured (that is, the desired balance of compression quality and speed), but also the platform type.

Table 6.8 describes the compression providers that each platform uses to compress data.

<table>
<thead>
<tr>
<th>BIG-IP Platform</th>
<th>Compression provider used</th>
</tr>
</thead>
<tbody>
<tr>
<td>6400/6800</td>
<td>If a hardware compression card is installed, the system uses the hardware card; if the hardware card is too busy or if no hardware card is installed, the system uses zlib.</td>
</tr>
<tr>
<td>8400</td>
<td>If a hardware compression card is installed, the system uses either the hardware card or zlib, depending on the gzip level you have configured; if no hardware card is installed, the system uses zlib.</td>
</tr>
<tr>
<td>8800</td>
<td>The system uses either the hardware card or zlib, depending on the gzip level you have configured; if the hardware card is too busy, the system uses zlib.</td>
</tr>
</tbody>
</table>

*Table 6.8 Use of compression providers on BIG-IP platforms*
Understanding gzip levels

To understand how adaptive compression operates, you must understand the gzip compression level setting in the HTTP profile. For more information about configuring an HTTP profile using the Configuration utility, including setting gzip compression levels, see the Configuration Guide for BIG-IP® Local Traffic Management.

When you enable adaptive compression, the system utilizes the gzip compression level that you set in the HTTP profile in different ways depending on which hardware compression provider the system contains. When you create an HTTP profile, you set a gzip compression level in the range of \( 9 - 0 \). The higher the gzip compression level, the better the quality of the compression, and therefore the more resources the system must use to reach that specified quality.

For example, you might set the gzip compression level to 9 if you are utilizing the BIG-IP system RAM cache feature to store response data. The reason for this is that the stored data in the RAM cache is continually re-used in responses, and therefore you want the quality of the compression of that data to be very high.

As the traffic flow on the BIG-IP system increases, the system automatically decreases the compression quality from the gzip compression level that you set in the profile. When the gzip compression level decreases to the point where the hardware compression provider is capable of providing the specified compression level, the system uses the hardware compression providers rather than the software compression providers to compress the HTTP server responses.

Specifying gzip levels on BIG-IP 6400, 6800, and 8400 systems

When you enable adaptive compression on a BIG-IP 6400, 6800, or 8400 system that contains a hardware card, the gzip compression level setting of the HTTP profile affects how the system performs compression.

Table 6.9, on page 6-27, shows the effect of gzip levels on the 6400, 6800, and 8400 platforms.
Specifying gzip levels on BIG-IP 8800 systems

When you enable adaptive compression on a BIG-IP 8800 system that contains a hardware card, the gzip compression level setting of the HTTP profile affects how the system performs compression.

Table 6.10, on page 6-28, shows the effect of gzip levels on an 8800 platform.
To configure adaptive compression, you not only configure the compression settings in a custom HTTP profile, but also use one of the command line interfaces. Using a command line interface is the way to change the compression strategy from Speed (the default strategy) to Adaptive.

Note that the Adaptive compression strategy uses the gzip Compression Level setting that you configure in the HTTP profile. For more information on gzip levels and the gzip Compression Level setting in particular, see Understanding gzip levels, on page 6-26 and Specifying a compression level, on page 6-20.

### Viewing compression statistics

You can use the bigpipe utility to view compression statistics for the BIG-IP system. You can view information about traffic throughput, as well as compression ratio totals for each hardware compression provider.

Table 6.11, on page 6-29, describes the compression statistics that are available.

---

### Table 6.10  Effect of gzip levels on 8800 systems

<table>
<thead>
<tr>
<th>gzip level</th>
<th>Effect on compression providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between 9 and 4, inclusive</td>
<td>When you set the gzip compression level to between 9 and 4, inclusive, the software compression provider performs the compression until the system load begins to increase. Then the BIG-IP system adaptively reduces the gzip level (that is, the quality of the compression). When the number of tasks handled by the software compression providers exceeds a certain value (defined by the bigdb key Compression.ProviderBusy), the hardware compression provider begins to compress the responses. When the number of tasks handled by the software compression providers falls below that same value, the software compression providers resume compressing the responses.</td>
</tr>
<tr>
<td>Between 3 and 0, inclusive</td>
<td>When you set the gzip compression level to between 3 and 0, inclusive, the hardware compression provider performs the compression until the number of tasks exceeds a certain value (defined by the bigdb key Compression.ProviderBusy). Then the software compression providers begin to compress the responses. When the number of tasks performed by the hardware compression providers falls below that same value, the hardware compression provider resumes compressing the responses. <strong>Note:</strong> The hardware provider does not support gzip levels 1 and 0. Therefore, if you specify 1 or 0, the hardware provider actually compresses the data at a level of slightly above 2. When the hardware compression provider becomes busy, the software providers can compress the data at level 1 or 0.</td>
</tr>
</tbody>
</table>

---

### Configuring adaptive compression

To configure adaptive compression, you not only configure the compression settings in a custom HTTP profile, but also use one of the command line interfaces. Using a command line interface is the way to change the compression strategy from Speed (the default strategy) to Adaptive.

Note that the Adaptive compression strategy uses the gzip Compression Level setting that you configure in the HTTP profile. For more information on gzip levels and the gzip Compression Level setting in particular, see Understanding gzip levels, on page 6-26 and Specifying a compression level, on page 6-20.
Figure 6.1 shows an example of the results of the `bigpipe` command `http show`. Note that any compression statistics appear in the last line of the output.

<table>
<thead>
<tr>
<th>Compression statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pre</td>
<td>Amount of uncompressed traffic that the BIG-IP system received.</td>
</tr>
<tr>
<td>post</td>
<td>Amount of compressed traffic that the BIG-IP system sent.</td>
</tr>
</tbody>
</table>
| null                  | Amount of uncompressed content that the BIG-IP system wrapped in compression headers.  
  Note: The system wraps content in compression headers, but does not compress the content when one of two situations occurs: either the system exceeds the amount of compression (in megabytes) for which it is licensed, or the CPU saver is active. |
| saved                 | Ratio between the amount of content before compression and the amount of content after compression. |

Table 6.11 Compression statistics described

GLOBAL HTTP STATISTICS
- requests (total, max in conn, GET, POST) = (0, 0, 0, 0)
- requests (v0.9, v1.0, v1.1) = (0, 0, 0)
- responses (v0.9, v1.0, v1.1) = (0, 0, 0)
- responses (2xx, 3xx, 4xx, 5xx) = (0, 0, 0, 0)
- response size ( <1k, 1-4k, 4-16k, 16-32k, 32-64k) = (0, 0, 0, 0, 0)
- Set-Cookie header insertions = 0
- RAM cache (entries, size, evict) = (0, 0, 0)
- RAM cache hit count (hits, misses, misses all) = (0, 0, 0)
- RAM cache byte count (hits, misses, misses all) = (0, 0, 0)

COMPRESSION STATISTICS --
- total bytes (pre, post, null, saved) = (0, 0, 0, 0.00%)

Figure 6.1 Sample output of the `http show` command on a BIG-IP 8800 system
Chapter 6

Configuring the RAM Cache feature

This section describes how to configure the properties of the RAM Cache feature on the BIG-IP system. A RAM cache is a cache of HTTP objects stored in the BIG-IP system’s RAM that are reused by subsequent connections to reduce the amount of load on the back-end servers.

Getting started with RAM caching

There are several concepts to consider before you configure the RAM Cache feature on the BIG-IP system.

• When to use the RAM cache
• What items you can cache
• The cache mechanism

When to use the RAM Cache feature

The RAM Cache feature provides the ability to reduce the traffic load to back-end servers. This ability is useful if an object on a site is under high demand, if the site has a large quantity of static content, or if the objects on the site are compressed.

◆ High-demand objects
  This feature is useful if a site has periods of high demand for specific content. With the RAM cache configured, the content server only has to serve the content to the BIG-IP system once per expiration period.

◆ Static content
  This feature is also useful if a site consists of a large quantity of static content such as CSS files, javascript files, or images and logos.

◆ Content compression
  For compressible data, the RAM cache can store data for clients that can accept compressed data. When used in conjunction with the compression feature on the BIG-IP system, the RAM cache takes stress off of the BIG-IP system and the content servers.

The items you can cache

The RAM Cache feature is fully compliant with the cache specifications described in RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1. This means you can configure the RAM Cache feature to cache the following content types:

• 200, 203, 206, 300, 301, and 410 responses
• Responses to GET methods, by default
• Other HTTP methods for URIs specified for inclusion in cached content, or specified in an iRule
• Content based on the User-Agent and Accept-Encoding values. The RAM cache holds different content for Vary headers.
The items that the RAM cache does not cache are:

- Private data specified by cache control headers
- HEAD, PUT, DELETE, TRACE, and CONNECT methods, by default

**Understanding the RAM Cache mechanism**

The default RAM cache configuration caches only responses to HTTP GET methods. However, you can use the RAM cache to cache other methods, too, including non-HTTP methods. You do this by specifying a URI in the URI Include or Pin list within an HTTP profile, or by writing an iRule.

The remainder of the section answers the following questions:

- Which responses does the BIG-IP system actually cache?
- What is the effect of adding URIs to the Include List in the HTTP profile?
- What is the order of preference that the BIG-IP system uses when processing the Pin List, Include List, and Exclude List?

**Which responses does the BIG-IP system cache?**

The BIG-IP system determines which responses to cache by evaluating both the HTTP request and the response. Table 6.12 shows the criteria within both a request and a response that the system uses before determining whether to include or exclude a response from the cache.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Action</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTTP Request Criteria</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The system invokes a CACHE::disable command within an HTTP_REQUEST event.</td>
<td>Exclude</td>
<td>Always</td>
</tr>
<tr>
<td>The request matches an item in the Exclude list of the HTTP profile.</td>
<td>Exclude</td>
<td>Always</td>
</tr>
<tr>
<td>The request matches an item in the Pin list of the HTTP profile.</td>
<td>Include</td>
<td>Always</td>
</tr>
<tr>
<td>The request matches an item in the Include list of the HTTP profile.</td>
<td>Include</td>
<td>Always</td>
</tr>
<tr>
<td>The request specifies a method other than GET, including non-standard HTTP methods.</td>
<td>Exclude</td>
<td>Always, except when overridden by an item in the Pin or Include lists, or by an iRule.</td>
</tr>
<tr>
<td>The BIG-IP system processes an iRule triggered by the CACHE_REQUEST event.</td>
<td>Include</td>
<td>Always</td>
</tr>
<tr>
<td><strong>HTTP Response Criteria</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 6.12 Request and response criteria for determining whether to cache a response*
The value of the `Vary` header consists of the `User-Agent` and/or `Accept-Encoding` request header only.

Include multiple response for a single URL

Always

The value of the `Vary` header changes in a subsequent response for the same resource.

Invalidates any existing cache entries

Always

The value of the `Vary` header consists of any other request headers or the special value * (asterisk)

Exclude

Always

The response includes a status code other than the following:
- 200 (OK)
- 203 (Non-Authoritative Information)
- 206 (Partial Content)
- 300 (Multiple Choices)
- 301 (Permanent Redirect)
- 410 (Gone)

Exclude

Always

The value of the `Content-Length` header in the response is 0.

Exclude

Always, except when the response has no `Content-Length` header, that is, with chunked `Transfer-Encoding`, or the response relies on connection termination by the server.

The object size of the response is outside the configured minimum and maximum response object size and the maximum cache size.

Exclude

Always, whenever the object size exceeds the maximum cache size.

Otherwise, always, except when overridden by an item in the `Pin` or `Include` lists, or by an iRule.

Note: The term `object size` refers to response headers as well as the body of the response.

The response contains an `Expires` header that has an invalid value.

Exclude

Always, except when overridden by an item in the `Pin` or `Include` lists, or by an iRule.

The request contains a query string, but the response contains no `Expires` header.

Exclude

Always, except when overridden by an item in the `Pin` or `Include` lists, or by an iRule.

The response contains a `Cache-Control` header with any of these values: `no-store`, `no-cache`, or `private`.

Exclude

Always, except when overridden by an item in the `Pin` or `Include` lists, or by an iRule.

The request contains an `Authorization` header, and the `Cache-Control` header in the response lacks any of these values: `s-maxage`, `must-revalidate`, `public`.

Exclude

Always, except when overridden by an item in the `Pin` or `Include` lists, or by an iRule.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Action</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>The value of the <code>Vary</code> header consists of the <code>User-Agent</code> and/or <code>Accept-Encoding</code> request header only.</td>
<td>Include multiple response for a single URL</td>
<td>Always</td>
</tr>
<tr>
<td>The value of the <code>Vary</code> header changes in a subsequent response for the same resource.</td>
<td>Invalidates any existing cache entries</td>
<td>Always</td>
</tr>
<tr>
<td>The value of the <code>Vary</code> header consists of any other request headers or the special value * (asterisk)</td>
<td>Exclude</td>
<td>Always</td>
</tr>
<tr>
<td>The response includes a status code other than the following: 200 (OK), 203 (Non-Authoritative Information), 206 (Partial Content), 300 (Multiple Choices), 301 (Permanent Redirect), 410 (Gone)</td>
<td>Exclude</td>
<td>Always</td>
</tr>
<tr>
<td>The value of the <code>Content-Length</code> header in the response is 0.</td>
<td>Exclude</td>
<td>Always, except when the response has no <code>Content-Length</code> header, that is, with chunked <code>Transfer-Encoding</code>, or the response relies on connection termination by the server.</td>
</tr>
<tr>
<td>The object size of the response is outside the configured minimum and maximum response object size and the maximum cache size. <strong>Note</strong>: The term <code>object size</code> refers to response headers as well as the body of the response.</td>
<td>Exclude</td>
<td>Always, whenever the object size exceeds the maximum cache size. Otherwise, always, except when overridden by an item in the <code>Pin</code> or <code>Include</code> lists, or by an iRule.</td>
</tr>
<tr>
<td>The response contains an <code>Expires</code> header that has an invalid value.</td>
<td>Exclude</td>
<td>Always, except when overridden by an item in the <code>Pin</code> or <code>Include</code> lists, or by an iRule.</td>
</tr>
<tr>
<td>The request contains a query string, but the response contains no <code>Expires</code> header.</td>
<td>Exclude</td>
<td>Always, except when overridden by an item in the <code>Pin</code> or <code>Include</code> lists, or by an iRule.</td>
</tr>
<tr>
<td>The response contains a <code>Cache-Control</code> header with any of these values: <code>no-store</code>, <code>no-cache</code>, or <code>private</code>.</td>
<td>Exclude</td>
<td>Always, except when overridden by an item in the <code>Pin</code> or <code>Include</code> lists, or by an iRule.</td>
</tr>
<tr>
<td>The request contains an <code>Authorization</code> header, and the <code>Cache-Control</code> header in the response lacks any of these values: <code>s-maxage</code>, <code>must-revalidate</code>, <code>public</code>.</td>
<td>Exclude</td>
<td>Always, except when overridden by an item in the <code>Pin</code> or <code>Include</code> lists, or by an iRule.</td>
</tr>
</tbody>
</table>

Table 6.12 Request and response criteria for determining whether to cache a response
What is the effect of adding URLs to the Include List of an HTTP profile?

For URIs in the Include List of an HTTP profile, the BIG-IP system caches responses for all request methods. All other constraints still apply.

What is the order of preference that the BIG-IP system uses?

When using URI lists to determine which response you want the BIG-IP system to cache, the system uses the following order of preference, in decreasing order:

- Exclude List
- Pin List
- Include List

Understanding BIG-IP system actions based on content

The BIG-IP system takes certain actions on cached content, depending on the type of content. Table 6.13 lists and describes these actions.

<table>
<thead>
<tr>
<th>Action</th>
<th>Type of Cached Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove</td>
<td>The BIG-IP system removes all cookie headers from cached content.</td>
</tr>
<tr>
<td>Modify</td>
<td>The BIG-IP system removes all hop-by-hop headers when served. These headers include:</td>
</tr>
<tr>
<td></td>
<td>Connection, Keep-Alive, and Transfer Encoding.</td>
</tr>
<tr>
<td>Add</td>
<td>The BIG-IP system adds a Date header to cached content that includes the current time on</td>
</tr>
<tr>
<td></td>
<td>the BIG-IP system. The system also adds an Age header to cached content that reflects</td>
</tr>
<tr>
<td></td>
<td>the amount of time that the item has been in the cache. Note that this setting is</td>
</tr>
<tr>
<td></td>
<td>enabled by default in the HTTP profile. You can disable this setting by disabling the</td>
</tr>
<tr>
<td></td>
<td>Insert Age Header in the profile.</td>
</tr>
<tr>
<td>Store as is</td>
<td>The BIG-IP system stores all other headers unchanged, as the system receives them.</td>
</tr>
</tbody>
</table>

Table 6.13  RAM cache action based on header type

Clearing items from the RAM cache

The RAM cache removes the least frequently used items in the cache. This prevents stale items from taking up room in the cache when newer items are selected for caching. The cache also uses a scoring system to remove stale items after a period of time. When a cached item reaches its age limit, the item expires and the system removes it from the cache. You can use the HTTP profile attributes to control the size of each cache instance and the rate at which the BIG-IP system removes expired items from the cache. For more information about these attributes, see the following section, Understanding RAM Cache settings, on page 6-34.
Understanding RAM Cache settings

To configure the RAM cache, you need to configure the RAM Cache settings of the HTTP profile. These settings provide the ability to turn on the cache and fine-tune it for a specific implementation. In addition to configuring the RAM cache objects in the HTTP profile, you may want to use bigpipe utility commands, configure bigdb configuration keys, or create iRules. For information on creating iRules for the RAM cache, see the F5 Networks web site http://devcentral.f5.com.

The default items stored by the cache are HTTP GET responses. However, you can specify URIs in the URI list if you want to cache POST and GET methods for a particular URI.

You can implement the RAM cache in any of these ways:

- You can create a custom profile based on the default http profile, and then modify the RAM Cache settings.
- You can implement the http-lan-optimized-caching profile as is. This profile contains settings that are already configured to optimize RAM caching.
- You can create a custom profile based on the http-lan-optimized-caching profile, and then modify the RAM Cache settings.

For more information, see Using an HTTP profile for RAM caching, following, and Using the http-lan-optimized-caching profile, on page 6-38.

Using an HTTP profile for RAM caching

As previously described, one way to implement the RAM cache is to create a custom profile based on the default http profile. Table 6.14 shows the RAM cache settings that you can specify within a custom HTTP profile that you create. Configuring these settings means either specifying a value where no default value exists, or changing a default value.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Cache</td>
<td>Specifies whether the RAM Cache feature is enabled or disabled.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Maximum Cache Size</td>
<td>Specifies the maximum size in megabytes for the RAM cache. When the cache reaches the maximum size, the system starts removing the oldest entries.</td>
<td>100</td>
</tr>
<tr>
<td>Maximum Entries</td>
<td>Specifies the maximum number of entries that can be in the RAM cache.</td>
<td>10000</td>
</tr>
</tbody>
</table>

Table 6.14  RAM cache settings of an HTTP profile
### Table 6.14  RAM cache settings of an HTTP profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Age</td>
<td>Specifies how long in seconds that the system considers the cached content to be valid.</td>
<td>3600</td>
</tr>
<tr>
<td>Minimum Object Size</td>
<td>Specifies the smallest object in bytes that the system considers eligible for caching.</td>
<td>500</td>
</tr>
<tr>
<td>Maximum Object Size</td>
<td>Specifies the largest object in bytes that the system considers eligible for caching.</td>
<td>50000</td>
</tr>
<tr>
<td>URI Caching</td>
<td>Specifies whether the system retains or excludes certain URIs in the RAM cache. The process forces the system either to cache URIs that typically are ineligible for caching, or to not cache URIs that typically are eligible for caching.</td>
<td>Not Configured</td>
</tr>
<tr>
<td>URI List</td>
<td>Specifies the Uniform Resource Identifiers (URIs) that the system either includes in or excludes from caching.</td>
<td>No default value</td>
</tr>
<tr>
<td>Pin List</td>
<td>Lists the URIs for responses that you want the system to store indefinitely in the RAM cache.</td>
<td></td>
</tr>
<tr>
<td>Include List</td>
<td>Lists the URIs that are typically ineligible for caching, but the system caches them. When you add URIs to the Include List, the system caches the GET methods and other methods, including non-HTTP methods.</td>
<td></td>
</tr>
<tr>
<td>Exclude List</td>
<td>Lists the URIs that are typically eligible for caching, but the system does not cache them.</td>
<td></td>
</tr>
<tr>
<td>Ignore Headers</td>
<td>Specifies how the system processes client-side Cache-Control headers when RAM caching is enabled.</td>
<td>All</td>
</tr>
<tr>
<td>All</td>
<td>Specifies that the system disregards all Cache-Control headers.</td>
<td></td>
</tr>
<tr>
<td>Cache-Control:max-age</td>
<td>Specifies that the system ignores only the Cache-Control:max-age header.</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>Specifies that the system honors all Cache-Control headers.</td>
<td></td>
</tr>
<tr>
<td>Insert Age Header</td>
<td>Specifies, when enabled, that the system inserts Date and Age headers in the cached entry. The Date header contains the current date and time on the BIG-IP system. The Age header contains the length of time the content has been in the cache.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Aging Rate</td>
<td>Specifies how quickly the system ages a cache entry. The aging rate ranges from 0 (slowest aging) to 10 (fastest aging).</td>
<td>9</td>
</tr>
</tbody>
</table>
Optimizing data compression and RAM caching

The BIG-IP system contains a set of custom HTTP profiles that F5 Networks has already created for you. The purpose of these profiles is to optimize the performance of data compression and RAM caching. Based on the http default profile, these F5-defined profiles are pre-configured specifically to ensure optimal compression and RAM cache performance. These profiles are:

- http-acceleration
- http-wan-optimized-compression
- http-lan-optimized-caching
- http-wan-optimized-compression-caching

Using the http-acceleration profile

The http-acceleration profile is an HTTP-type profile. This profile is effectively a custom profile that the BIG-IP system has already created for you. By implementing the http-acceleration profile, you can accelerate HTTP traffic, without having to create a custom profile to do so.

In some cases, you can simply enable the http-acceleration profile from within another BIG-IP system module, such as the BIG-IP WebAccelerator system.

The http-acceleration profile inherits its settings and their default values from the http profile, but some of the setting values have been changed. This profile is a good choice to use when you want to optimize HTTP acceleration, while still using the default values from the http profile for the other configuration settings.

You can use the http-acceleration profile as is, or you can create another custom profile, specifying the http-acceleration profile as the parent profile.

Most of the default values of the http-acceleration profile are the same as the default values of the http profile. However, a few default values are different, and these are listed in Table 6.15.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Cache</td>
<td>Specifies whether the RAM Cache feature is enabled or disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Minimum Object Size</td>
<td>Specifies the smallest object in bytes that the system considers eligible for caching.</td>
<td>0</td>
</tr>
<tr>
<td>Maximum Object Size</td>
<td>Specifies the largest object in bytes that the system considers eligible for caching.</td>
<td>4194304</td>
</tr>
</tbody>
</table>

Table 6.15 Values of an http-acclerator profile that differ from those of the http profile
Using the http-wan-optimized-compression profile

The `http-wan-optimized-compression` profile is an HTTP-type profile. This profile is effectively a custom profile that the BIG-IP system has already created for you. By implementing the `http-wan-optimized-compression` profile, you can optimize the performance of data compression, without having to create a custom profile to do so.

The `http-wan-optimized-compression` profile inherits its settings and their default values from the `http` profile, but some of the setting values have been changed. This profile is a good choice to use when you want to optimize data compression, while still using the default values from the `http` profile for the other configuration settings.

You can use the `http-wan-optimized-compression` profile as is, or you can create another custom profile, specifying the `http-wan-optimized-compression` profile as the parent profile.

Most of the default values of the `http-wan-optimized-compression` profile are the same as the default values of the `http` profile. However, a few default values are different, and these are listed in Table 6.16.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>Enables or disables the HTTP compression feature.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Compression Buffer Size</td>
<td>Specifies the maximum number of compressed bytes that the BIG-IP system buffers before deciding whether or not to insert a <code>Content-Length</code> header into the response that specifies the compressed size.</td>
<td>131072</td>
</tr>
<tr>
<td>HTTP/1.0 Requests</td>
<td>Enables or disables compression of responses to HTTP/1.0 client requests.</td>
<td>Enabled</td>
</tr>
<tr>
<td>gzip Compression Level</td>
<td>Specifies the amount and rate of compression.</td>
<td>1 - Least Compression (Fastest)</td>
</tr>
<tr>
<td>gzip Memory Level</td>
<td>Specifies the number of kilobytes of memory that the BIG-IP system uses for internal compression buffers when compressing a server response.</td>
<td>16</td>
</tr>
<tr>
<td>gzip Window Size</td>
<td>Specifies the number of kilobytes in the window size that the BIG-IP system uses when compressing a server response.</td>
<td>64</td>
</tr>
<tr>
<td>Vary Header</td>
<td>Enables or disables the insertion of a <code>Vary</code> header into cacheable server responses.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Response Chunking</td>
<td>Specifies how to handle chunking for HTTP responses. Possible values are <code>Unchunk</code>, <code>Rechunk</code>, <code>Selective</code>, and <code>Preserve</code>.</td>
<td>Selective</td>
</tr>
</tbody>
</table>

Table 6.16 Values of an `http-wan-optimized-compression` profile that differ from those of the `http` profile
Using the http-lan-optimized-caching profile

The **http-lan-optimized-caching** profile is an HTTP-type profile. This profile is effectively a custom profile that the BIG-IP system has already created for you. By implementing the **http-lan-optimized-caching** profile, you can optimize the performance of RAM caching, without having to create a custom profile to do so.

The **http-lan-optimized-caching** profile inherits its settings and their default values from the **http** profile, but some of the setting values have been changed. This profile is a good choice to use when you want to optimize RAM caching, while still using the default values from the **http** profile for the other configuration settings.

You can use the **http-lan-optimized-caching** profile as is, or you can create another custom profile, specifying the **http-lan-optimized-caching** profile as the parent profile.

Most of the default values of the **http-lan-optimized-caching** profile are the same as the default values of the **http** profile. However, a few default values are different, and these are listed in Table 6.17.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Cache</td>
<td>Specifies whether the RAM Cache feature is enabled or disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Maximum Cache Size</td>
<td>Specifies the maximum size in megabytes for the RAM cache. When the cache reaches the maximum size, the system starts removing the oldest entries.</td>
<td>10</td>
</tr>
<tr>
<td>Maximum Age</td>
<td>Specifies how long in seconds that the system considers the cached content to be valid.</td>
<td>86400</td>
</tr>
<tr>
<td>Minimum Object Size</td>
<td>Specifies the smallest object in bytes that the system considers eligible for caching.</td>
<td>0</td>
</tr>
<tr>
<td>Maximum Object Size</td>
<td>Specifies the largest object in bytes that the system considers eligible for caching.</td>
<td>2000000</td>
</tr>
</tbody>
</table>

*Table 6.17 Values of an **http-lan-optimized-caching** profile that differ from those of the **http** profile*
Using the http-wan-optimized-compression-caching profile

The **http-wan-optimized-compression-caching** profile is another type of HTTP profile. The **http-wan-optimized-compression-caching** profile combines the default configuration values of the **http**, **http-wan-optimized-compression**, and **http-lan-optimized-caching** profiles. This profile is a good choice to use when you want to optimize both data compression and RAM caching, while still using the default values from the **http** profile for the other configuration settings.

For information on the default configuration values for this profile, see *Understanding HTTP profile settings*, on page 6-2, *Using the http-wan-optimized-compression profile*, on page 6-37, and *Using the http-lan-optimized-caching profile*, on page 6-38.
Configuring FTP profile settings

The BIG-IP system includes a profile type that you can use to manage File Transfer Protocol (FTP) traffic. 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 5, Understanding Profiles.

Table 6.18 lists these configurable settings, along with a short description of each, and the default values. Following this table are descriptions of specific settings.

<table>
<thead>
<tr>
<th>General property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the user-supplied name of the profile. Specifying a name for your profile is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which your custom profile is derived.</td>
<td>ftp</td>
</tr>
<tr>
<td>Translate Extended</td>
<td>Ensures compatibility between IP version 4 and IP version 6 clients and servers when using the FTP protocol.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Data Port</td>
<td>Allows the FTP service to run on an alternate port.</td>
<td>20</td>
</tr>
<tr>
<td>Protocol Security</td>
<td>Specifies, when checked (enabled), that the system inspects FTP traffic for security vulnerabilities, by using a security profile in BIG-IP® Protocol Security Module. This option is available only when you are licensed for the module.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

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 (IPv4) and IP version 6 (IPv6) systems. For example, if a client system running IPv4 sends the FTP PASV command to a server running IPv6, the BIG-IP system automatically translates the PASV command to the equivalent FTP command for IPv6 systems, EPSV.

The BIG-IP system translates the FTP commands EPRV and PORT in the same way.

It is highly unlikely that you will 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 IPv4 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.

Enabling security for FTP traffic

When the BIG-IP system includes a license for the BIG-IP Application Security Manager, you can enable a security scan for FTP traffic by using the Advanced Firewall feature in the application security configuration. For more information, see Working with the Advanced Firewall, in the Configuration Guide for BIG-IP® Application Security Management.
Configuring SIP profile settings

The BIG-IP system includes a services profile that you can use to manage Session Initiation Protocol (SIP) traffic. Session Initiation Protocol is an application-layer protocol that manages sessions consisting of multiple participants, thus enabling real-time messaging, voice, data, and video. A session can be a simple two-way telephone call or Instant Message dialogue, or a complex, collaborative, multi-media conference call that includes voice, data, and video.

SIP sessions, which are application level sessions, run through one of three Layer 4 protocols: SCTP, TCP, or UDP. The SIP profile configures how the system handles SIP sessions. The specified Layer 4 protocol profile configures the virtual server to open the required port to allow data to flow through the BIG-IP system. When you assign a SIP profile to a virtual server, you can also assign either an SCTP, TCP, or UDP profile to the server. If you do not assign one of these protocol profiles to the server, the BIG-IP system automatically assigns one for you. For more information about protocol profiles, see Chapter 8, Managing Protocol Profiles.

The SIP profile automatically configures the BIG-IP system to handle persistence for SIP sessions using Call-ID. The Call-ID is a globally unique identifier that groups together a series of messages, which are sent between communicating applications. You can customize how the system handles persistence for SIP sessions. To do this, you create a SIP persistence profile. In order to use a SIP persistence profile, you must also use a SIP profile. You assign both the SIP profile and the SIP persistence profile to the same virtual server. For more information about configuring SIP persistence, see Understanding SIP persistence profile settings, on page 7-18.

Understanding SIP profile settings

On the New SIP Profile screen in the Configuration utility, the SIP configuration settings are organized into two categories: General Properties, and Settings. When you create a SIP profile, you can implement it as is, or modify the settings. You can also modify the profile at a later date. For specific procedures on configuring a profile, see Chapter 5, Understanding Profiles.

Table 6.19, on page 6-43, lists the configurable settings in a SIP profile, along with a short description of each setting and the default value. Following this table are more detailed descriptions of the settings and the procedures for changing them.
Before configuring a SIP profile, it is helpful to have a description of certain settings that you might want to change.

<table>
<thead>
<tr>
<th>General property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the user-supplied name of the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which your custom profile is derived.</td>
<td>sip</td>
</tr>
<tr>
<td>Maximum Size (Bytes)</td>
<td>Specifies the maximum SIP message size that the BIG-IP system accepts. The system drops the connection if the SIP message is larger than this value.</td>
<td>65535 bytes</td>
</tr>
<tr>
<td>Dialog Aware</td>
<td>Specifies that you want the BIG-IP system to snoop SIP dialog information and forward it to a known SIP dialog. Enabling this setting causes the Community setting to appear on the screen.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Community</td>
<td>Specifies a string to indicate the proxy functional group to which this profile belongs. This setting appears on the screen only when you have enabled the Dialog Aware setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Terminate on BYE</td>
<td>Enables or disables the closing of a connection when a BYE transaction finishes. <strong>A BYE transaction</strong> is a message that an application sends to another application when it is ready to close the connection between the two. When the BIG-IP system encounters a BYE transaction, if the Terminate on BYE setting is enabled, then the system ends the connection. Use this parameter with UDP connections only, not with TCP connections.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Insert Via Header</td>
<td>Enables or disables the insertion of a Via header into a SIP request. A <strong>Via header</strong> indicates where the message originated. The response message uses this routing information.</td>
<td>Disabled</td>
</tr>
<tr>
<td>User Via</td>
<td>When the Insert Via Header setting is enabled, specifies a string that you want the system to insert as the value for the top Via header in a SIP request. This value specifies the SIP protocol, as well as a virtual address and port.</td>
<td>No default value</td>
</tr>
<tr>
<td>Secure Via Header</td>
<td>Enables or disables the insertion of a Secure Via header into a SIP request. A <strong>Secure Via Header</strong> indicates where the message originated. Use this parameter with SSL/TLS to secure the system. The response message uses this routing information.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Insert Record-Route Header</td>
<td>Enables or disables the insertion of a Record-Route SIP header, which indicates the next hop for the following SIP request messages.</td>
<td>Disabled (Unchecked)</td>
</tr>
</tbody>
</table>

*Table 6.19 Configuration settings of a SIP profile*
Specifying a profile name

To create a SIP profile, you must specify a unique name for the profile. The **Name** setting is the only setting for which you must actively specify a value when creating a SIP 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 **sip** profile as the parent profile, or you can use another SIP profile that you have already created.

To specify a parent profile, locate the **Parent Profile** setting and select a profile name.

Specifying maximum message size

The BIG-IP system accepts incoming SIP messages that are 65535 bytes or smaller. If a SIP message exceeds this value, the system drops the connection.

You can change the maximum message size that the system accepts by typing a different value in the **Maximum Size (Bytes)** box.

Enabling dialog snooping

When you enable the **Dialog Aware** setting, the BIG-IP system snoops SIP dialog information and automatically forwards SIP messages to the known SIP dialog. Enabling this setting causes the Configuration utility to display the **Community** setting.

Specifying a community string

With the **Community** setting, you can specify the name of a proxy functional group. You use this setting in the case where you need multiple virtual servers, each referencing a SIP-type profile, and you want more than one of those profiles to belong to the same proxy functional group. This setting only appears on the screen when you have enabled the **Dialog Aware** setting.

Specifying connection termination criteria

The BIG-IP system terminates a SIP connection when either the application that initiated the session (client) or the application that answered the initiated session (server) issues a BYE transaction. This is appropriate when a SIP session is running on UDP. However, if you are running SIP on a SCTP or TCP connection, you must disable the **Terminate on BYE** setting.

To disable the **Terminate on BYE** setting, click the Custom box, and then clear the **Enabled** box.
Handling SIP headers

An optional feature in a SIP profile is header insertion. You can specify whether the BIG-IP system inserts headers into SIP requests. Specifically, you can enable or disable insertion of Via headers, Secure Via headers, and Record-Route headers. When you assign the configured SIP profile to a virtual server, the BIG-IP system then inserts the header specified in the profile into any SIP request that the BIG-IP system sends to a pool or pool member.

Inserting a Via header into a SIP message

You can configure the BIG-IP system to insert a Via header into SIP requests to indicate where a SIP message originated. The SIP responses use this routing information to locate the session initiator.

By default, the Insert Via Header setting is disabled. To enable this setting, click the Custom box.

Specifying a user value

When you enable the Insert Via Header setting, you then specify a User Via value. The User Via value is a string that the system inserts as the value of the top Via header value in a SIP request. The string specifies the SIP protocol, as well as a virtual address and port. An example of a User Via string is SIP/2.0/UDP 10.10.10.10:5060.

Inserting a Secure Via header into a SIP message

When you are using SSL/TLS (over TCP) to create a secure channel with the server node, you can configure the BIG-IP system to insert a Secure Via header into a SIP request. The Secure Via Header setting indicates where the SIP message originated.

By default, the Secure Via Header setting is disabled. To enable this setting, click the Custom box.

Inserting a Record-Route header into a SIP message

You can also configure the BIG-IP system to insert a Record-Route header into SIP requests. The Insert Record-Route Header setting specifies the next hop for the following SIP requests.

By default, the Insert Record-Route Header setting is disabled. To enable this setting, click the Custom box.
Customizing SIP persistence

You can customize how the BIG-IP system handles persistence for SIP sessions. You do this by creating a specific type of persistence profile called a SIP persistence profile. It is important to note that you must always create and use a SIP services profile (as described in Understanding SIP profile settings, on page 6-42) when using a SIP persistence profile. To do this, you create both profiles, and then assign the profiles to the same virtual server. For more information about the SIP persistence profile, see SIP persistence, on page 7-18.
Configuring RTSP profile settings

The BIG-IP system includes a profile type that you can use to manage Real Time Streaming Protocol (RTSP) traffic. Real Time Streaming Protocol (RTSP) is a protocol used for streaming-media presentations. Using RTSP, a client system can control a remote streaming-media server and allow time-based access to files on a server.

The RTSP profile in the BIG-IP system supports these features:

- The setup of streaming media over UDP. In this case, the control connection opens the required ports to allow data to flow through the BIG-IP system.
- Interleaved data over the control connection, essentially streaming media over TCP.
- Real Networks tunneling of RTSP over HTTP, through the RTSP port (554).

When you assign an RTSP profile to a virtual server, you should also assign a TCP profile. If you do not assign a TCP profile, the BIG-IP system automatically assigns one for you.

Table 6.20 lists the configurable settings in an RTSP profile, along with a short description of each setting and the default values.

<table>
<thead>
<tr>
<th>General property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the user-supplied name of the profile. A name for your profile is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which your custom profile is derived.</td>
<td>rtsp</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the number of seconds that a UDP connection is idle before the connection is eligible for deletion.</td>
<td>300</td>
</tr>
<tr>
<td>Maximum Header Size</td>
<td>Specifies the largest RTSP request or response header, in bytes, that the BIG-IP system allows before terminating the connection.</td>
<td>4096</td>
</tr>
<tr>
<td>Maximum Queued Data</td>
<td>Specifies the maximum amount of data, in bytes, that the BIG-IP system buffers, before considering the connection to be unusable and subsequently terminating the connection.</td>
<td>32768</td>
</tr>
<tr>
<td>Unicast Redirect</td>
<td>If you are using unicast streams and you enable this setting, the client is able to specify the destination address and port for the streamed data. For security reasons however, the destination address used is the source of the request.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Multicast Redirect</td>
<td>If you are using multicast streams and you enable this setting, the client has permission to supply a different destination for the streamed data.</td>
<td>Disabled (Unchecked)</td>
</tr>
</tbody>
</table>

Table 6.20  Configuration settings of an RTSP profile
A common configuration for the RTSP profile is one that includes RTSP clients and media servers, as well as RTSP proxies to manage accounting and authentication tasks. In this proxied configuration, you most likely want the streaming media from the servers to pass directly to the client, bypassing the RTSP proxy servers.

To implement this configuration, you configure the BIG-IP system by creating two virtual servers, one for processing traffic to and from the external network, and one for processing traffic to and from the internal network. For each virtual server, you assign a separate RTSP profile.

<table>
<thead>
<tr>
<th>General property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Reconnect</td>
<td>When this setting is enabled and a disconnected control connection has been resumed, the BIG-IP system persists the resumed control connection to the correct server. Typical clients do not support this behavior.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Real HTTP Persistence</td>
<td>When this setting is enabled, the system automatically persists Real Networks-tunneled RTSP data over HTTP, which is over the RTSP port. The default value is Enabled. When you disable this setting, a user can override the default behavior with an iRule.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Check Source</td>
<td>When checked (enabled), specifies that the system examines the origin of the message to determine whether the message came from the client or the server.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Proxy</td>
<td>Specifies whether the RTSP profile is associated with an RTSP proxy configuration. Possible values are: None, External, and Internal.</td>
<td>None</td>
</tr>
<tr>
<td>Proxy Header</td>
<td>Specifies a name for the header that the system inserts into a SETUP request. The name must begin with the string X-. The value of this header typically consists of information about the client IP address and is read by another RTSP profile. Note that the system removes this header from the request prior to sending the request to the server for processing.</td>
<td>No default value</td>
</tr>
<tr>
<td>RTP Port</td>
<td>Specifies the port number that a Microsoft® Media Services server uses. Normally, Microsoft Media Services uses a fixed Realtime Transport Protocol (RTP) port number. With this setting, you can specify a port number instead of using the fixed RTP port number.</td>
<td>0</td>
</tr>
<tr>
<td>RTCP Port</td>
<td>Specifies an RTCP port companion of the RTP Port value. Normally, Microsoft Media Services uses a fixed Real-Time Control Protocol (RTCP) port number. With this setting, you can specify a port number instead of using the fixed RTCP port number.</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.20 Configuration settings of an RTSP profile
With this configuration:

- The RTSP profile on the external virtual server passes client IP address information to the RTSP profile on the internal virtual server.
- The RTSP profile on the internal virtual server extracts the client IP address information from the request, processes the media server’s response, and opens the specified ports on the BIG-IP system. Opening these ports allows the streaming media to bypass the RTSP proxy servers as the data travels from the server to the client.

The client IP address information is stored in the **Proxy Header** setting that you specify in the RTSP profile.
Configuring iSession profile settings

The BIG-IP system includes the iSession profile type, which creates an optimization tunnel between two BIG-IP systems that are separated by a wide area network. More specifically, an iSession profile causes the system to:

- Hold enqueued data
- Compress a certain type and amount of data, based on the ingress wire speed

Table 6.21 lists the configurable settings in an iSession profile, along with a short description of each setting and the default values.

<table>
<thead>
<tr>
<th>General property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the user-supplied name of the profile. A name for your profile is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which your custom profile is derived.</td>
<td>isession</td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies that the system uses this profile to optimize traffic over the WAN.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Compression</td>
<td>Specifies the method the system uses for symmetric compression:</td>
<td>deflate</td>
</tr>
<tr>
<td></td>
<td>off: Specifies that the system does not use compression.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deflate: Specifies that the system uses the deflate data compression algorithm.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lzo: Specifies that the system uses the Lempel-Ziv-Oberhumer (lzo) data compression algorithm.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bzip2: Specifies that the system uses the bzip2 data compression algorithm.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>adaptive: Specifies that the system uses the compression algorithm that is the most suitable for the current traffic.</td>
<td></td>
</tr>
<tr>
<td>Port Transparency</td>
<td>Specifies that the destination port specified by the client is preserved over the WAN.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Reuse Connection</td>
<td>Specifies that the system saves and reuses connections between the local and remote WAN Optimization Modules.</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

Table 6.21 Configuration settings of an iSession profile
For the complete procedure on implementing a Local Traffic Manager paired-tunneling configuration using an iSession profile, see the guide titled *BIG-IP® Local Traffic Manager: Implementations*. 

![Table 6.21: Configuration settings of an iSession profile](image)
Enabling Session Persistence

- Introducing session persistence
- Persistence types and their profiles
Introducing session persistence

Using the BIG-IP® local traffic management system, 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 several 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, or in a cookie on the client’s machine. 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 7-4.
- To understand profiles in general, see Chapter 5, 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).

You can also use an iRule to enable persistence for SSL-terminated requests, that is, requests that the BIG-IP system terminates by performing decryption and re-encryption and by handling SSL certificate authentication. In this type of iRule, you can use an HTTP header insertion iRule command to insert an SSL session ID as a header into an HTTP request.

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 17, Writing iRules.

Using the OneConnect profile with session persistence

When you configure session persistence, the BIG-IP system tracks and stores session data, such as the pool member that serviced a client request. Configuring a persistence profile for a virtual server ensures that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions.

The Request-URI header in an HTTP request stores certain session data. Occasionally, however, for Cookie and Universal persistence types specifically, the BIG-IP system ignores the session data in this header, and sends requests to an unexpected node. For example, this issue can occur when clients send requests to a virtual server through an internet proxy device. You can prevent this problem by creating a OneConnect profile, and assigning both the OneConnect profile and the persistence profile to the virtual server.

The following two sections explain the effect of a OneConnect profile on session persistence.

HTTP parsing without a OneConnect profile

If the virtual server does not reference a OneConnect profile, the BIG-IP system performs load balancing for each TCP connection. Once the TCP connection is load balanced, the system sends all requests that are part of the connection to the same pool member.

For example, if the virtual server does not reference a OneConnect profile, and the BIG-IP system initially sends a client request to node A in pool A, the system inserts a cookie for node A. Then, within the same TCP connection, if the BIG-IP system receives a subsequent request that contains a cookie for node B in pool B, the system ignores the cookie information and incorrectly sends the request to node A instead.
HTTP parsing using a OneConnect profile

Using a OneConnect type of profile solves the problem. If the virtual server references a OneConnect profile, the BIG-IP system can perform load balancing for each request within the TCP connection. That is, when an HTTP client sends multiple requests within a single connection, the BIG-IP system is able to process each HTTP request individually. The BIG-IP system sends the HTTP requests to different destination servers if necessary.

For example, if the virtual server references a OneConnect profile and the client request is initially sent to node A in pool A, the BIG-IP system inserts a cookie for node A. Then, within the same TCP connection, if the BIG-IP system receives a subsequent request that contains a cookie for node B in pool B, the system uses that cookie information and correctly sends the request to node B.

Troubleshooting tips

To mitigate issues when the BIG-IP system ignores persistence information in the Request-URI header and therefore sends requests to an incorrect node, you can take these actions:

- Associate a OneConnect profile with the virtual server.
- Verify that the OneConnect Transformations setting in the HTTP profile is enabled.

For information on configuring a OneConnect profile and an HTTP profile, see the Configuration Guide for BIG-IP® Local Traffic Management.
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 5, Understanding Profiles.

Types of persistence

The persistence types that you can enable using a persistence profile are:

- **Cookie persistence**
  Cookie persistence uses an HTTP cookie stored on a client’s computer to allow the client to reconnect to the same server previously visited at a web site.

- **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.

- **Hash persistence**
  Hash persistence allows you to create a persistence hash based on an existing iRule.

- **Microsoft® Remote Desktop Protocol persistence**
  Microsoft® Remote Desktop Protocol (MSRDP) persistence tracks sessions between clients and servers running the Microsoft® Remote Desktop Protocol (RDP) service.

- **SIP persistence**
  SIP persistence is a type of persistence used for servers that receive Session Initiation Protocol (SIP) messages sent through UDP, SCTP, or TCP.

- **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 persistence**
  SSL persistence is a type of persistence that tracks non-terminated SSL sessions, using the SSL session ID. To enable persistence for terminated SSL sessions, see Chapter 9, Managing SSL Traffic, Chapter 17, Writing iRules, and the F5 Networks DevCentral web site, http://devcentral.f5.com.

- **Universal persistence**
  Universal persistence allows you to write an expression that defines what to persist on in a packet. The expression, written using the same expression syntax that you use in iRules™, defines some sequence of bytes to use as a session identifier.
Enabling Session Persistence

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 node 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 SSL persistence, should include the same node addresses in the virtual server mappings.

Note

With respect to Cookie profiles, this setting applies to the Cookie Hash method only.

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 node. Connection requests from the client that do not use persistence are load balanced according to the currently selected load balancing method.

Note

With respect to Cookie profiles, this setting applies to the Cookie Hash method only.

WARNING

In order for this setting to be effective, virtual servers that use pools with TCP or SSL 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).

WARNING

Enabling this setting can cause the BIG-IP system to direct client traffic to a pool other than that specified by the virtual server.

With respect to Cookie profiles, this setting applies to the Cookie Hash method only.
Cookie persistence

You can set up the BIG-IP system to use HTTP cookie persistence. Cookie persistence uses an HTTP cookie stored on a client’s computer to allow the client to reconnect to the same pool member previously visited at a web site.

There are four methods of cookie persistence available:

- HTTP Cookie Insert method
- HTTP Cookie Rewrite method
- HTTP Cookie Passive method
- Cookie Hash method

The method you choose to use affects how the cookie is handled by the BIG-IP system when it is returned to the client.

◆ Note

F5 recommends that you configure a OneConnect profile in addition to the Cookie profile, to ensure that the BIG-IP system load balances HTTP requests correctly. For more information, see Using the OneConnect profile with session persistence, on page 7-2.

Understanding Cookie profile settings

To implement cookie persistence, you can either use the default cookie profile, or create a custom profile. Table 7.1 shows the settings and values that make up a Cookie profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence. This setting is required.</td>
<td>Cookie</td>
</tr>
<tr>
<td>Cookie Method</td>
<td>Specifies the type of cookie processing that the BIG-IP system is to use.</td>
<td>HTTP Cookie Insert</td>
</tr>
<tr>
<td>Cookie Name</td>
<td>Specifies the name of the cookie that the BIG-IP system should look for or insert.</td>
<td>This value is autogenerated based on the pool name.</td>
</tr>
<tr>
<td>Expiration</td>
<td>Sets the expiration time of the cookie. Applies to the HTTP Cookie Insert and HTTP Cookie Rewrite methods only. When using the default (checked), the system uses the expiration time specified in the session cookie.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Hash Offset</td>
<td>With respect to Cookie persistence, this setting applies to the Cookie Hash method only.</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.1 Settings of a Cookie persistence profile
You can configure the **Cookie Method** setting in a cookie profile to one of four values.

### HTTP Cookie Insert method

If you specify **HTTP Cookie Insert** method within the profile, the information about the server to which the client connects is inserted in the header of the HTTP response from the server as a cookie. The cookie is named `BIGipServer<pool_name>`, and it includes the address and port of
the server handling the connection. The expiration date for the cookie is set based on the timeout configured on the BIG-IP system. **HTTP Cookie Insert** is the default value for the **Cookie Method** setting.

Note that the profile settings **Mirror Persistence**, **Match Across Services**, **Match Across Virtual Servers**, and **Match Across Pools** do not apply to the **HTTP Cookie Insert** method. These settings apply to the **Cookie Hash** method only.

**Tip**

*You can assign this type of profile to a Performance (HTTP) type of virtual server.*

**HTTP Cookie Rewrite method**

If you specify **HTTP Cookie Rewrite** method, the BIG-IP system intercepts a **Set-Cookie** header, named **BIGipCookie**, sent from the server to the client, and overwrites the name and value of the cookie. The new cookie is named **BIGipServer<pool_name>** and it includes the address and port of the server handling the connection.

**Important**

*We recommend that you use this method instead of the **HTTP Cookie Passive** method whenever possible.*

The **HTTP Cookie Rewrite** method requires you to set up the cookie created by the server. For the **HTTP Cookie Rewrite** method to succeed, there needs to be a blank cookie coming from the web server for the BIG-IP system to rewrite. With Apache variants, the cookie can be added to every web page header by adding the following entry to the **httpd.conf** file:

```
Header add Set-Cookie BIGipCookie=0000000000000000000000000...
```

(The cookie must contain a total of 120 zeros.)

**Note**

*For backward compatibility, the blank cookie can contain only 75 zeros. However, cookies of this size do not allow you to use iRules™ and persistence together.*

Note that the profile settings **Mirror Persistence**, **Match Across Services**, **Match Across Virtual Servers**, and **Match Across Pools** do not apply to the **HTTP Cookie Rewrite** method. These settings apply to the **Cookie Hash** method only.
HTTP Cookie Passive method

If you specify the HTTP Cookie Passive method, the BIG-IP system does not insert or search for blank Set-Cookie headers in the response from the server. This method does not try to set up the cookie. With this method, the server provides the cookie, formatted with the correct server information and timeout.

Important

We recommend that you use the HTTP Cookie Rewrite method instead of the HTTP Cookie Passive method whenever possible.

For the HTTP Cookie Passive method to succeed, there needs to be a cookie coming from the web server with the appropriate server information in the cookie. Using the Configuration utility, you generate a template for the cookie string, with encoding automatically added, and then edit the template to create the actual cookie.

For example, the following string is a generated cookie template with the encoding automatically added, where [pool name] is the name of the pool that contains the server, 336260299 is the encoded server address, and 20480 is the encoded port:

```
Set-Cookie:BIGipServer[poolname]=336268299.20480.0000; expires=Sat, 01-Jan-2002 00:00:00 GMT; path=/
```

To create your cookie from this template, type the actual pool name and an expiration date and time.

Alternatively, you can perform the encoding using the following equation for address (a.b.c.d):

```
d*(256^3) + c*(256^2) + b*256 +a
```

The way to encode the port is to take the two bytes that store the port and reverse them. Thus, port 80 becomes 80 * 256 + 0 = 20480. Port 1433 (instead of 5 * 256 + 153) becomes 153 * 256 + 5 = 39173.

With Apache variants, the cookie can be added to every web page header by adding the following entry to the httpd.conf file:

```
Header add Set-Cookie: "BIGipServer my_pool=184658624.20480.000; expires=Sat, 19-Aug-2002 19:35:45 GMT; path=/"
```

Note that the profile settings Mirror Persistence, Match Across Services, Match Across Virtual Servers, and Match Across Pools do not apply to the HTTP Cookie Passive method. These settings apply to the Cookie Hash method only.

Cookie Hash method

If you specify the Cookie Hash method, the hash method consistently maps a cookie value to a specific node. When the client returns to the site, the BIG-IP system uses the cookie information to return the client to a given node. With this method, the web server must generate the cookie; the BIG-IP system does not create the cookie automatically as it does when you use the HTTP Cookie Insert method.
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 7.2 shows the settings and their values that make up a Destination Address Affinity profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence profile. This setting is required.</td>
<td>Destination Address Affinity</td>
</tr>
<tr>
<td>Mirror Persistence</td>
<td>Specifies, when enabled (checked), that if the active unit goes into the standby mode, the system mirrors any persistence records to its peer.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Mask</td>
<td>Specifies the mask that the BIG-IP system should use before matching with an existing persistence entry.</td>
<td>255.255.255.255</td>
</tr>
</tbody>
</table>

*Table 7.2 Settings of a Destination Address Affinity persistence profile*
Hash persistence

Hash persistence allows you to create a persistence hash based on an existing iRule that uses the persist uie iRule command. Using hash persistence is the same as using universal persistence, except that with hash persistence, the resulting persistence key is a hash of the data, rather than the data itself.

Figure 7.1 shows an example of a iRule that implements hash persistence.

```
rule my_persist_irule {
    when HTTP_REQUEST {
        persist uie [HTTP::header myheader]
    }
}
```

Figure 7.1 Sample iRule for hash persistence

After implementing an iRule for hash persistence, you can type the command bigpipe persist show all, which shows that the system uses the hash of the data, rather than the data itself.

For example, if you have an iRule that specifies data strings data1 and data2, the bigpipe persist show all command shows the hash values of these strings (2356372769 and 1996459178, respectively) as the persistence values. Figure 7.2 shows this output.

```
# b persist show all
PERSISTENT CONNECTIONS --
Mode: hash Value: 2356372769
Virtual: 10.1.1.41:http Node: 10.10.1.190:http Age: 2sec
Mode: hash Value: 1996459178
Virtual: 10.1.1.41:http Node: 10.10.1.180:http Age: 2sec
```

Figure 7.2 Output showing use of hash values for persistence

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout</td>
<td>Specifies the duration, in seconds, of a persistence entry. For background information on setting timeout values, see Chapter 1, Introducing Local Traffic Management. Possible values are: Specify: Specifies the number of seconds before the persistence entry expires Indefinite: Specifies that the persistence entry does not expire.</td>
<td>180</td>
</tr>
<tr>
<td>Override Connection Limit</td>
<td>Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

Table 7.2 Settings of a Destination Address Affinity persistence profile
Note that if you use hash persistence and the BIG-IP system cannot find an entry in the persistence table for a connection, and the system has not yet chosen a pool member due to fallback persistence, then the system uses the hash value, rather than the specified load balancing method, to select the pool member.

To illustrate using the previous example, if the persistence table contains no entry for the hash value 2356372769, and the number of active nodes in the pool remains the same, then a session with that hash value for persistence is always persisted to node 10.10.10.190 (assuming that the node is active).

Understanding Hash profile settings

To implement hash persistence, you either use the default hash profile or create a custom profile. Table 7.3 shows the settings and their values that make up a Hash profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence profile. This setting is required.</td>
<td>Hash</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Hash Algorithm</td>
<td>Specifies the algorithm the system uses for hash persistence load balancing. The hash result is the input for the algorithm. Possible settings are:</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>• Default: Specifies that the system uses the index of pool members to obtain the hash result for the input to the algorithm.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CARP: Specifies that the system uses the Cache Array Routing Protocol (CARP) to obtain the hash result for the input to the algorithm.</td>
<td></td>
</tr>
<tr>
<td>Hash Offset</td>
<td>Specifies the start offset within the packet from which the system begins the hash when performing hash persistence load balancing. The default value of 0 (zero) indicates no offset.</td>
<td>0</td>
</tr>
<tr>
<td>Hash Length</td>
<td>Specifies the length of data within the packet, in bytes, that the system uses to calculate the hash value when performing hash persistence load balancing.</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.3 Settings of a Hash persistence profile
Microsoft Remote Desktop Protocol persistence

MSRDP persistence provides an efficient way of load balancing traffic and maintaining persistent sessions between Windows® clients and servers that are running the Microsoft® Remote Desktop Protocol (RDP) service. The recommended scenario for enabling MSRDP persistence feature is to create a load balancing pool that consists of members running Windows Server 2003 or Windows Server 2008, where all members belong to a Windows cluster and participate in a Windows session directory.

Benefits of MSRDP persistence

Normally, Windows servers running Microsoft Terminal Services can use a session broker (known as Terminal Services Session Directory in Windows Server 2003 and TS Session Broker in Windows Server 2008) to ensure that

---

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash Start Pattern</td>
<td>Specifies the string expression (Tcl regex) that describes the starting location of the hash pattern that the system uses to perform hash persistence load balancing.</td>
<td>No default value</td>
</tr>
<tr>
<td>Hash End Pattern</td>
<td>Specifies the string expression (Tcl regex) that describes the ending location of the hash pattern that the system uses to perform hash persistence load balancing.</td>
<td>No default value</td>
</tr>
<tr>
<td>Hash Buffer Limit</td>
<td>Specifies the maximum amount of data within which the system searches to locate the hashing pattern.</td>
<td>0</td>
</tr>
<tr>
<td>Hash More</td>
<td>Specifies that the system performs another hash operation after the current hash operation completes.</td>
<td>Disabled (unchecked)</td>
</tr>
<tr>
<td>iRule</td>
<td>Specifies an iRule to run that determines the persistence entry.</td>
<td>None</td>
</tr>
<tr>
<td>Timeout</td>
<td>The setting specifies the duration, in seconds, of a persistence entry. For background information on setting timeout values, see Chapter 1, Introducing Local Traffic Management. Possible values are: Specify: Specifies the number of seconds before the persistence entry expires. Indefinite: Specifies that the persistence entry does not expire.</td>
<td>180</td>
</tr>
<tr>
<td>Override Connection Limit</td>
<td>Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

*Table 7.3 Settings of a Hash persistence profile*
user sessions are assigned to specific servers. If a client initiates a connection request to the wrong terminal server, that server redirects the client to the appropriate server.

When you have a BIG-IP system, however, the incorrect server needs to redirect the client to the BIG-IP system virtual server, rather than to an individual server in the load balancing pool. To ensure that this happens, you can configure an MSRDP profile. With an MSRDP profile, the BIG-IP system uses a token that the session broker provides to maintain persistence records. If a user initiates a session for which no session broker token exists, the BIG-IP system makes load balancing decisions according to whichever load balancing method is configured for the pool.

In summary, using the BIG-IP system with an MSRDP persistence profile, in conjunction with a session broker, allows for higher scalability and a greater range and flexibility of load balancing options than when using a session broker alone.

Server Platform issues

By default, the BIG-IP system with MSRDP persistence enabled load balances connections according to the way that the user has configured the BIG-IP system for load balancing, as long as the session broker is configured on each server in the pool. Terminal Services Session Directory and TS Session Broker are features that are only available on Windows Server 2003 or Windows Server 2008 respectively. Therefore, each server in the pool must be a Windows Server 2003 or Windows Server 2008 server, if you want to use MSRDP persistence in default mode. Also, each client system must be running the remote desktop client software that is included with any Windows Server 2003 or Windows Server 2008 system.

If, however, you want to enable MSRDP persistence but your server platforms are running older versions of Windows (on which Session Directory or TS Session Broker is not available), you can enable MSRDP persistence in non-default mode. This causes the BIG-IP system to connect a client to the same Windows server by way of the user name that the client provides. Note that enabling MSRDP persistence in non-default mode (that is, with no session broker available on the servers) is less preferable than the default mode, because it provides limited load-balancing and redirection capabilities.

Configuring MSRDP persistence with a session broker

To enable MSRDP persistence in the default mode, you must configure a session broker on each Windows server in your load balancing pool. In addition to configuring a session broker, you must perform other Windows configuration tasks on those servers. However, before you configure your Windows servers, you must configure the BIG-IP system, by performing tasks such as creating a load-balancing pool and designating your Windows servers as members of that pool.

The following two sections describe BIG-IP system configuration tasks that are required to enable MSRDP persistence in default mode for a Windows client-server configuration using RDP.
Configuring MSRDP persistence without a session broker

When a server has no session broker, the server cannot share sessions with other servers, and therefore cannot perform any redirections when a connection to a server becomes disconnected. In lieu of session sharing, Windows clients provide data, in the form of a user name, to the BIG-IP system to allow the BIG-IP system to consistently connect that client to the same server. Enabling MSRDP persistence to behave in this way is the non-default mode.

Understanding MSRDP profile settings

To implement MSRDP persistence, you either use the default `msrdp` profile or create a custom profile. Table 7.4 shows the settings and their values that make up an MSRDP type of profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence profile. This setting is required.</td>
<td>Microsoft Remote Desktop</td>
</tr>
<tr>
<td>Mirror Persistence</td>
<td>Specifies, when enabled (checked), that if the active unit goes into the standby mode, the system mirrors any persistence records to its peer.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Timeout</td>
<td>Specifies the duration, in seconds, of a persistence entry. For background information on setting timeout values, see Chapter 1, <em>Introducing Local Traffic Management</em>. Possible values are: <strong>Specify</strong>: Specifies the number of seconds before the persistence entry expires. <strong>Indefinite</strong>: Specifies that the persistence entry does not expire.</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 7.4 Settings of an MSRDP persistence profile
Enabling Session Persistence

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has Session Directory</td>
<td>Specifies whether the server is running Session Directory.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Override Connection Limit</td>
<td>Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

*Table 7.4 Settings of an MSRDP persistence profile*
SIP persistence

Session Initiation Protocol is an application-layer protocol that manages sessions consisting of multiple participants, thus enabling real-time messaging, voice, data, and video. A session can be a simple two-way telephone call or Instant Message dialogue, or a complex, collaborative, multi-media conference call that includes voice, data, and video. With SIP, applications can communicate with one another by exchanging messages through the SCTP, TCP or UDP protocols.

SIP persistence is a type of persistence available for server pools. You can configure SIP persistence for proxy servers that receive SIP messages sent through the UDP profile. The BIG-IP system currently supports persistence for SIP messages sent through the UDP, TCP, or SCTP protocols.

Understanding SIP persistence profile settings

To implement SIP persistence, you either use the default Persistence Type, SIP, or create a custom profile. Table 7.5 shows the settings and values that make up a SIP persistence profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence profile. This setting is required.</td>
<td>SIP</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which your custom profile is derived.</td>
<td>sip_info</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node. Persistence across all virtual servers causes the traffic management system to maintain persistence for all connections requested by the same client, regardless of which virtual server hosts each individual connection initiated by the client.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry. Persistence across all pools causes the traffic management system to maintain persistence for all connections requested by the same client, regardless of which pool hosts each individual connection initiated by the client.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

Table 7.5 Settings of a SIP persistence profile
Enabling Session Persistence

The timeout value that you specify allows the BIG-IP system to free up resources associated with old SIP persistence entries, without having to test each inbound packet for one of the different types of SIP final messages. The default timeout value is 180 seconds. If you change the timeout value, we recommend that the value be no lower than the default.

**Important**

*For virtual servers processing UDP traffic, always check that the value of the SIP profile Timeout setting is at least as long (in seconds) as the value of the Idle Timeout setting of the UDP profile. Doing so ensures that SIP traffic is persisted properly.*
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.

You might want to use source address affinity persistence and SSL persistence together. In situations where an SSL session ID times out, or where a returning client does not provide a session ID, you may want the BIG-IP system to direct the client to the original pool member based on the client’s IP address. As long as the client’s source address affinity persistence record has not timed out, the BIG-IP system can successfully return the client to the appropriate 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 7.6 shows the settings and values that make up a Source Address Affinity profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence profile. This setting is required.</td>
<td>Source Address Affinity</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

Table 7.6 Settings of a Source Address Affinity persistence profile
Enabling Session Persistence

SSL persistence

SSL persistence is a type of persistence that tracks SSL sessions using the SSL session ID, and it is a property of each individual pool. Using SSL persistence can be particularly important if your clients typically have translated IP addresses or dynamic IP addresses, such as those that Internet service providers typically assign. Even when the client’s IP address changes, the BIG-IP system still recognizes the session as being persistent based on the session ID.

You might want to use SSL persistence and source address affinity persistence together. In situations where an SSL session ID times out, or where a returning client does not provide a session ID, you might want the BIG-IP system to direct the client to the original node based on the client’s IP address. As long as the client’s simple persistence record has not timed out, the BIG-IP system can successfully return the client to the appropriate node.

◆ Note

The SSL persistence type is only valid for systems that are not performing SSL certificate-based authentication of client requests or server responses. If you are using Client SSL or Server SSL profiles to configure certificate-based authentication, do not configure an SSL persistence profile. Instead, create an iRule to perform SSL session persistence.

### Table 7.6 Settings of a Source Address Affinity persistence profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout</td>
<td>Specifies the duration, in seconds, of a persistence entry. For background information on setting timeout values, see Chapter 1, Introducing Local Traffic Management. Possible values are: Specify: Specifies the number of seconds before the persistence entry expires. Indefinite: Specifies that the persistence entry does not expire.</td>
<td>180</td>
</tr>
<tr>
<td>Mask</td>
<td>Specifies the mask that the BIG-IP system should use before matching with an existing persistence entry.</td>
<td>None</td>
</tr>
<tr>
<td>Map Proxies</td>
<td>Enables or disables proxy mapping.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Override Connection Limit</td>
<td>Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>
The SSL persistence profile

To implement SSL persistence, you either use the default \texttt{ssl} profile or create a custom profile. Table 7.7 shows the settings and their values that make up an SSL persistence profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence profile. This setting is required.</td>
<td>SSL</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Timeout</td>
<td>Specifies the number of seconds before a persistence entry times out. That is, this setting sets the SSL session ID timeout value, which determines how long the BIG-IP system stores a given SSL session ID before removing the ID from the system. For background information on setting timeout values, see Chapter 1, \textit{Introducing Local Traffic Management}.</td>
<td>300</td>
</tr>
<tr>
<td>Override Connection Limit</td>
<td>Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

\textit{Table 7.7 Settings of an SSL persistence profile}

Universal persistence

Included in the BIG-IP system’s Universal Inspection Engine (UIE) is a set of functions that you can specify within BIG-IP system iRules to direct traffic in more granular ways. Using these iRule functions, you can write expressions that direct traffic based on content data, or direct traffic to a specific member of a pool.

\textit{Universal persistence} takes this iRules feature one step further, by allowing you to use the iRule \texttt{persist uie} command to implement persistence for sessions based on content data, or based on connections to a specific member of a pool. Universal persistence does this by defining some sequence of bytes to use as a session identifier.
To use iRule expressions for persistence, a universal persistence profile includes a setting that specifies the name of the iRule containing the expression.

Figure 7.3 shows an example of a iRule that implements universal persistence.

```plaintext
rule my_persist_irule {
  when HTTP_REQUEST { persist uie [HTTP::header myheader] } }
```

**Figure 7.3 Sample iRule for universal persistence**

Unlike hash persistence, which uses a hash of the data as the persistence key, universal persistence uses the data itself as the persistence key.

Once you have created an iRule and specified the iRule name within a universal profile, you must assign both the iRule and the profile to the appropriate virtual server as resources.

For more information on iRules, see the F5 Networks DevCentral web site [http://devcentral.f5.com](http://devcentral.f5.com), and Chapter 17, Writing iRules.

**Note**

F5 recommends that you configure a OneConnect profile in addition to the Universal profile, to ensure that the BIG-IP system load balances HTTP requests correctly. For more information, see Using the OneConnect profile with session persistence, on page 7-2.

### The universal persistence profile

To implement universal persistence, you can either use the default universal profile or create a custom profile. Table 7.8 shows the settings and values that make up a universal persistence profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Persistence Type</td>
<td>Specifies the type of persistence. This setting is required.</td>
<td>Universal</td>
</tr>
<tr>
<td>Match Across Services</td>
<td>Specifies that all persistent connections from a client IP address that go to the same virtual IP address also go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Match Across Virtual Servers</td>
<td>Specifies that all persistent connections from the same client IP address go to the same node.</td>
<td>Disabled (Cleared)</td>
</tr>
</tbody>
</table>

**Table 7.8 Settings of a universal persistence profile**
Match Across Pools Specifies that the BIG-IP system can use any pool that contains this persistence entry.

iRule Specifies the name of an existing iRule that the BIG-IP system should run to determine a persistence entry.

Timeout Specifies the duration, in seconds, of a persistence entry. For background information on setting timeout values, see Chapter 1, Introducing Local Traffic Management. Possible values are:

- **Specify**: Specifies the number of seconds before the persistence entry expires.
- **Indefinite**: Specifies that the persistence entry does not expire.

Override Connection Limit Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.

Table 7.8 Settings of a universal persistence profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match Across Pools</td>
<td>Specifies that the BIG-IP system can use any pool that contains this persistence entry.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>iRule</td>
<td>Specifies the name of an existing iRule that the BIG-IP system should run to determine a persistence entry.</td>
<td>None</td>
</tr>
</tbody>
</table>
| Timeout                 | Specifies the duration, in seconds, of a persistence entry. For background information on setting timeout values, see Chapter 1, Introducing Local Traffic Management. Possible values are:
- **Specify**: Specifies the number of seconds before the persistence entry expires.
- **Indefinite**: Specifies that the persistence entry does not expire. | 180                 |
| Override Connection Limit| Specifies, when checked (enabled), that the system allows you to specify that pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden. | Disabled (Cleared)  |
Managing Protocol Profiles

- Introducing protocol profiles
- Configuring a Fast L4 profile
- Configuring a Fast HTTP profile
- Configuring an HTTP Class profile
- Configuring TCP profiles
- Configuring a UDP profile
- Configuring an SCTP profile
Introducing protocol profiles

Some of the BIG-IP® system profiles that you can configure are known as protocol profiles. The protocol profiles types are:

- Fast L4
- Fast HTTP
- HTTP Class
- TCP
- UDP
- SCTP

For each protocol 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 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, Fast HTTP, HTTP Class, TCP, UDP, and SCTP profiles. For information on configuring other types of profiles, see the following chapters:

- For information on the HTTP, FTP, RTSP, and SIP profiles, see Chapter 6, Managing Application Layer Traffic.
- For information on the profiles used for session persistence, see Chapter 7, Enabling Session Persistence.
- For information on the Client and Server SSL profiles, see Chapter 9, Managing SSL Traffic.
- For information on the profiles used for authenticating traffic through a remote authentication server, see Chapter 10, Authenticating Application Traffic.
- For information on the OneConnect, NTLM, Statistics, and Stream profiles, see Chapter 11, Using Additional Profiles.
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).

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 8.1 lists and describes the settings of a Fast L4 profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This setting specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>fastL4</td>
</tr>
<tr>
<td>Reset on Timeout</td>
<td>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.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Reassemble IP Fragments</td>
<td>If this setting is enabled, the BIG-IP system reassembles IP fragments.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>This setting specifies the number of seconds that a connection is idle before the connection is eligible for deletion. For background information on setting idle timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 8.1 Settings of a Fast L4 profile
### Table 8.1 Settings of a Fast L4 profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Handshake Timeout</td>
<td><strong>Specify:</strong> 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. <strong>Disabled:</strong> Specifies that the system does not apply a timeout to a TCP handshake. <strong>Indefinite:</strong> Specifies that the acceptable duration for a TCP handshake is indefinite.</td>
<td>5</td>
</tr>
<tr>
<td>Max Segment Size Override</td>
<td>Overrides the maximum segment size (MSS), which is 1460. Possible values are: <strong>Disabled:</strong> Specifies that you want the maximum segment size to remain at 1460. <strong>Specify:</strong> 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).</td>
<td>Disabled</td>
</tr>
<tr>
<td>PVA Acceleration</td>
<td>This setting specifies the maximum acceleration mode that you prefer the system to use. Note that depending on the virtual server configuration, the system might or might not accelerate traffic in this mode. (For more information, see Configuring PVA hardware acceleration, on page 8-5.) Possible values are Full, Assisted, or None. Additional information on this setting follows this table.</td>
<td>Full</td>
</tr>
<tr>
<td>IP ToS to Client</td>
<td>This setting specifies the Type of Service level that the BIG-IP system assigns to IP packets when sending them to clients.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>IP ToS to Server</td>
<td>This setting specifies the Type of Service level that the BIG-IP system assigns to IP packets when sending them to servers.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>Link QoS to Client</td>
<td>This setting specifies the Quality of Service level that the BIG-IP system assigns to IP packets when sending them to clients.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>Link QoS to Server</td>
<td>This setting specifies the Quality of Service level that the BIG-IP system assigns to IP packets when sending them to servers.</td>
<td>Pass Through</td>
</tr>
<tr>
<td>TCP Timestamp Mode</td>
<td>Specifies the action that the BIG-IP system should take on TCP timestamps. Possible values are: Preserve, Strip, and Rewrite.</td>
<td>Preserve</td>
</tr>
<tr>
<td>TCP Window Scale Mode</td>
<td>Specifies the action that the BIG-IP system should take on TCP windows. Possible values are: Preserve and Strip.</td>
<td>Preserve</td>
</tr>
<tr>
<td>Generate Internal Sequence Numbers</td>
<td>Enables the BIG-IP system to generate its own sequence numbers for SYN packets, according to RFC 1948.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
### Table 8.1 Settings of a Fast L4 profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strip Sack OK</td>
<td>Enables the BIG-IP system to block a TCP SackOK option from passing to the server on an initiating SYN.</td>
<td>Disabled</td>
</tr>
<tr>
<td>RTT from Client</td>
<td>Specifies that the BIG-IP system should use TCP timestamp options to measure the round-trip time to the client.</td>
<td>Disabled</td>
</tr>
<tr>
<td>RTT from Server</td>
<td>Specifies that the BIG-IP system should use TCP timestamp options to measure the round-trip time to the server.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Loose Initiation</td>
<td>Specifies, when checked (enabled), that the system initializes a connection when it receives any TCP packet, rather that 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. <strong>Important:</strong> Enabling loose initiation can permit stray packets to pass through the system. This can pose a security risk and reduce system performance.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Loose Close</td>
<td>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.</td>
<td>Disabled</td>
</tr>
<tr>
<td>TCP Close Timeout</td>
<td>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.</td>
<td>5</td>
</tr>
<tr>
<td>Hardware SYN Cookie Protection</td>
<td>Enables or disables hardware SYN cookie protection when PVA10 is present on the system. This feature is available on certain hardware platforms only.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Software SYN Cookie Protection</td>
<td>Enables or disables software SYN cookie protection when PVA10 is not present on the system.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
Configuring PVA hardware acceleration

Once you implement a Fast L4 profile, the BIG-IP system automatically selects the most efficient PVA hardware acceleration mode for Layer 4 traffic. Possible modes are **Full**, **Assisted**, and **None**.

The particular hardware acceleration mode that the BIG-IP system selects depends on these factors:

- **The Fast L4 profile settings**
  The mode that the BIG-IP selects is influenced by the way that you configure the settings of the Fast L4 profile.

- **The virtual server configuration**
  The mode that the BIG-IP system selects is influenced by the specific features that you assigned to the virtual server (such as pools, SNAT pools, and iRules).

- **A monitor assigned to associated nodes**
  For full PVA acceleration, you must assign monitors to the relevant nodes.

- **The value of the PVA Acceleration setting**
  The **PVA Acceleration** setting in the Fast L4 profile defines the maximum amount of hardware acceleration that you want to allow, for Layer 4 traffic passing through the virtual server. Therefore, if you set the value to:

  - **Full**: The system can set hardware acceleration to any of the three modes (**Full**, **Assisted**, or **None**), depending on the virtual server configuration. This is the default value.
  
  - **Assisted**: The system can set hardware acceleration to either Assisted or **None** mode, depending on the virtual server configuration.

  - **None**: The system does not perform hardware acceleration.

One reason that you might want to set the maximum hardware acceleration setting to less than **Full** is for viewing connections with the **bigpipe conn show** command. This command only shows Layer 4 connections when the hardware acceleration mode is set to **Assisted** or **None**. If the mode is set to **Full**, the **bigpipe conn show** command shows no Layer 4 connections.

Depending on the current mode to which hardware acceleration is automatically set, the BIG-IP system accelerates Layer 4 traffic as described in Table 8.2, on page 8-6.

---

**Important**

*If you have a VLAN group configured on the BIG-IP system and its Transparency Mode setting is set to Translucent or Transparent, the BIG-IP system automatically sets the PVA Acceleration value to None.*
Hardware Acceleration Mode | Result
---|---
**Full**<br>The hardware acceleration processes all Layer 4 traffic. Layer 4 traffic is **not** managed through the use of BIG-IP software features. In this case, the BIG-IP system treats client-side and server-side packets as part of the same connection.<br><br>An example of using hardware acceleration in **Full** mode is when you want to load balance Layer 4 traffic to two servers, using the Round Robin load balancing method, with no session persistence or iRules.

**Assisted**<br>The BIG-IP system load balances all SYN packets, while the hardware acceleration assists with the remaining packets, including the tearing down of connections.<br><br>An example of using hardware acceleration in **Assisted** mode is when you want to load balance Layer 4 traffic using a dynamic load balancing method, or using a simple iRule that examines the IP addresses contained in the packets.<br><br>**Note:** When the BIG-IP system sets the hardware acceleration mode to **Assisted**, a Fast L4 profile is compatible with SNATs and SNAT pools, as well as with source address affinity persistence.

**None**<br>The hardware acceleration does not process any Layer 4 traffic. Instead, the BIG-IP application manages all Layer 4 traffic.<br><br>An example of using hardware acceleration in **None** mode is when you want to load balance traffic using an HTTP profile, as well as an iRule that performs delayed binding and cookie session persistence.

**Table 8.2** Effect of PVA hardware acceleration mode on Layer 4 traffic
Configuring a Fast HTTP profile

The Fast HTTP profile is a configuration tool designed to speed up certain types of HTTP connections. This profile combines selected features from the TCP, HTTP, and OneConnect profiles into a single profile that is optimized for the best possible network performance. When you associate this profile with a virtual server, the virtual server processes traffic packet-by-packet, and at a significantly higher speed.

You might consider using a Fast HTTP profile when:

- You do not need features such as session persistence, remote server authentication, SSL traffic management, and TCP optimizations, nor HTTP features such as data compression, pipelining, and RAM Cache.
- You do not need to maintain source IP addresses.
- You want to reduce the number of connections that are opened to the destination servers.
- The destination servers support connection persistence, that is, HTTP/1.1, or HTTP/1.0 with **Keep-Alive** headers. Note that IIS servers support connection persistence by default.
- You need basic iRule support only (such as limited Layer 4 support and limited HTTP header operations). For example, you can use the iRule events **CLIENT_ACCEPTED**, **SERVER_CONNECTED**, and **HTTP_REQUEST**.

A significant benefit of using a Fast HTTP profile is the way in which the profile supports connection persistence. Using a Fast HTTP profile ensures that for client requests, the BIG-IP system can transform or add an HTTP **Connection** header to keep connections open. Using the profile also ensures that the BIG-IP system pools any open server-side connections. This support for connection persistence can greatly reduce the load on destination servers by removing much of the overhead caused by the opening and closing of connections. For more information on HTTP header transformation, see Chapter 6, *Managing Application Layer Traffic*. For more information on the pooling of server-side connections, see Chapter 11, *Using Additional Profiles*.

◆ **Note**

*The Fast HTTP profile is incompatible with all other profile types. Also, you cannot use this profile type in conjunction with VLAN groups, or with the IPv6 address format.*
Understanding Fast HTTP profile settings

You can use the default `fasthttp` profile as is, or create a custom Fast HTTP profile. Table 8.3 lists and describes the settings of the Fast HTTP profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td><code>fasthttp</code></td>
</tr>
<tr>
<td>Reset on Timeout</td>
<td>Specifies, when checked (enabled), that the system sends a TCP RESET packet when a connection times out, and deletes the connection.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>This setting specifies the number of seconds that a connection is idle before the connection flow is eligible for deletion because it has no traffic. Possible values are: Specify, Immediate, and Indefinite. For background information on setting idle timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
<tr>
<td>Maximum Segment Size</td>
<td>Specifies a maximum segment size (MSS) override for server-side connections. The default setting is 0, which corresponds to an MSS of 1460. To override this size, you can specify any integer between 536 and 1460.</td>
<td>0</td>
</tr>
<tr>
<td>Client Close Timeout</td>
<td>Specifies the number of seconds after which the system closes a client connection, when the system either receives a client FIN packet or sends a FIN packet to the client. This setting overrides the Idle Timeout setting. Possible values are: Specify, Immediate, and Indefinite. For more information, see the online help.</td>
<td>5</td>
</tr>
<tr>
<td>Server Close Timeout</td>
<td>Specifies the number of seconds after which the system closes a client connection, when the system either receives a server FIN packet or sends a FIN packet to the server. This setting overrides the Idle Timeout setting. Possible values are: Specify, Immediate, and Indefinite. For more information, see the online help.</td>
<td>5</td>
</tr>
<tr>
<td>Unclean Shutdown</td>
<td>Specifies how the system handles closing connections. Possible values are: Disabled, Enabled, and Fast. For more information, see the online help.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Force HTTP 1.0 Response</td>
<td>Specifies, when checked (enabled), that the server sends responses to clients in the HTTP/1.0 format. This effectively disables client chunking and pipelining.</td>
<td>Disabled ( Cleared)</td>
</tr>
</tbody>
</table>

Table 8.3 Settings of a Fast HTTP profile
### Maximum Pool Size
Specifies the maximum number of connections a load balancing pool can accept. A setting of 0 specifies that there is no maximum; that is, a pool can accept an unlimited number of connections.

### Minimum Pool Size
Specifies the minimum number of connections that a load balancing pool can accept. A setting of 0 specifies that there is no minimum.

### Ramp-Up Increment
Specifies the increment in which the system makes additional connections available, when all available connections are in use.

### Maximum Reuse
Specifies the maximum number of times that the system can re-use a current connection.

### Idle Timeout Override
Specifies the number of seconds after which a server-side connection in a pool is eligible for deletion, when the connection has no traffic. This setting overrides the Idle Timeout setting. Possible values are: Specify, Disabled, and Indefinite. For more information, see the online help.

### Replenish
Specifies whether the BIG-IP system should maintain a steady-state maximum number of back-end connections. If you disable this setting, the system does not keep a steady-state maximum of connections to the back end, unless the number of connections to the pool drops below the value specified in the Minimum Pool Size setting.

### Parse Requests
Specifies, when checked (enabled), that the system parses the HTTP data in the connection stream. Note that if you are using a Fast HTTP profile for non-HTTP traffic, you should disable this setting to shield against dynamic denial-of-service (DDOS) attacks.

### Maximum Header Size
Specifies the maximum amount of HTTP header data that the system buffers before making a load balancing decision.

### Maximum Requests
Specifies the maximum number of requests that the system allows for a single client-side 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 of 0 means that the system allows an infinite number of requests per client-side connection.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Pool Size</td>
<td>Specifies the maximum number of connections a load balancing pool can accept. A setting of 0 specifies that there is no maximum; that is, a pool can accept an unlimited number of connections.</td>
<td>2048</td>
</tr>
<tr>
<td>Minimum Pool Size</td>
<td>Specifies the minimum number of connections that a load balancing pool can accept. A setting of 0 specifies that there is no minimum.</td>
<td>0</td>
</tr>
<tr>
<td>Ramp-Up Increment</td>
<td>Specifies the increment in which the system makes additional connections available, when all available connections are in use.</td>
<td>4</td>
</tr>
<tr>
<td>Maximum Reuse</td>
<td>Specifies the maximum number of times that the system can re-use a current connection.</td>
<td>0</td>
</tr>
<tr>
<td>Idle Timeout Override</td>
<td>Specifies the number of seconds after which a server-side connection in a pool is eligible for deletion, when the connection has no traffic. This setting overrides the Idle Timeout setting. Possible values are: Specify, Disabled, and Indefinite. For more information, see the online help.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Replenish</td>
<td>Specifies whether the BIG-IP system should maintain a steady-state maximum number of back-end connections. If you disable this setting, the system does not keep a steady-state maximum of connections to the back end, unless the number of connections to the pool drops below the value specified in the Minimum Pool Size setting.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Parse Requests</td>
<td>Specifies, when checked (enabled), that the system parses the HTTP data in the connection stream. Note that if you are using a Fast HTTP profile for non-HTTP traffic, you should disable this setting to shield against dynamic denial-of-service (DDOS) attacks.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Maximum Header Size</td>
<td>Specifies the maximum amount of HTTP header data that the system buffers before making a load balancing decision.</td>
<td>32768</td>
</tr>
<tr>
<td>Maximum Requests</td>
<td>Specifies the maximum number of requests that the system allows for a single client-side connection. When the specified limit is reached, the final response contains a <em>Connection: close</em> header is followed by the closing of the connection. The default setting of 0 means that the system allows an infinite number of requests per client-side connection.</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 8.3 Settings of a Fast HTTP profile*
When writing iRules, you can specify a number of events and commands that the Fast HTTP profile supports. For more information about these iRule events and commands, see the DevCentral web site http://devcentral.f5.com, as well as Chapter 17, Writing iRules.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert XForwarded For</td>
<td>Specifies whether the system inserts the XForwarded For: header in an HTTP request with the client IP address, to use with connection pooling. Possible settings are Enabled and Disabled. For more information, see the online help.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Request Header Insert</td>
<td>Specifies a string that the system inserts as a header in an HTTP request. If the header exists already, the system does not replace it.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

Table 8.3 Settings of a Fast HTTP profile
Configuring an HTTP Class profile

An HTTP Class profile is a configuration tool that you can use to classify HTTP traffic. When you classify traffic, you forward traffic to a destination based on an examination of traffic headers or content. Use of an HTTP Class profile is an efficient way for the BIG-IP system to classify traffic based on criteria that you specify. Although you can perform these same traffic-classification functions using the iRules feature, using an HTTP Class profile simplifies this process.

The destination you specify can be either a load balancing pool or a URL. To classify HTTP traffic, you configure an HTTP Class profile to specify strings that match a list type. The list types that you can use for string matching are:

- Host names
- URIs
- Headers
- Cookies

The string that you can match to one of these lists can be either a pattern string or a regular expression.

Note that list types are case-sensitive for pattern strings. For example, the system treats the pattern string `www.f5.com` differently from the pattern string `www.F5.com`. You can override this case-sensitivity by using the Linux `regexp` command.

Once the BIG-IP system matches the string to the corresponding list type, the system can send the traffic to a pool that you specify. Alternatively, you can create an HTTP Class profile that forwards a client request from the targeted HTTP virtual server to an HTTPS virtual server instead of to a pool.

Understanding HTTP Class profile settings

You can use the default `httpclass` profile as is, or create a custom HTTP Class profile. Table 8.4, on page 8-12, lists and describes the settings of an HTTP Class profile.
### Table 8.4 Settings of an HTTP Class profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>httpclass</td>
</tr>
<tr>
<td>Application Security</td>
<td>Specifies that you want a virtual server to forward traffic to the Application Security Manager application. In this case, the HTTP Class profile is the equivalent of an Application Security Manager application security class. This setting appears only when Application Security Manager is licensed on the BIG-IP system.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>WebAccelerator</td>
<td>Specifies that you want a virtual server to forward traffic to the WebAccelerator application. This setting appears only when WebAccelerator is licensed on the BIG-IP system.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Hosts</td>
<td>Specifies whether the host names used as criteria for routing HTTP requests constitute all hosts or individual hosts that you specify. A value of <strong>Match All</strong> directs the system to forward HTTP requests from all hosts. A value of <strong>Match Only</strong> directs the system to forward HTTP requests based on only those hosts you specify.</td>
<td>Match All</td>
</tr>
<tr>
<td>Host List</td>
<td>Specifies individual host names to be used as criteria for routing HTTP requests. Using the <strong>Entry Type</strong> list, you must also identify each host name as either a pattern string or a regular expression. This setting appears only when the value of Hosts is <strong>Match Only</strong>.</td>
<td>No default value</td>
</tr>
<tr>
<td>URI Paths</td>
<td>Specifies whether the URIs used as criteria for routing HTTP requests constitute all URIs or individual URIs that you specify. A value of <strong>Match All</strong> directs the system to forward HTTP requests from all URIs. A value of <strong>Match Only</strong> directs the system to forward HTTP requests based on only those URIs you specify.</td>
<td>Match All</td>
</tr>
<tr>
<td>URI List</td>
<td>Specifies individual URI paths to be used as criteria for routing HTTP requests. Using the <strong>Entry Type</strong> list, you must also identify each URI as either a pattern string or a regular expression. This setting appears only when the value of URI Paths is <strong>Match Only</strong>.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

*Note: When you use pattern strings, this list type is case-sensitive. For more information, see Configuring an HTTP Class profile, on page 8-11.*
### Table 8.4 Settings of an HTTP Class profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>Specifies whether the headers and their values, used as criteria for routing HTTP requests constitute all headers or individual headers that you specify. A value of <strong>Match All</strong> directs the system to forward HTTP requests based on all headers. A value of <strong>Match Only</strong> directs the system to forward HTTP requests based on only those headers you specify.</td>
<td><strong>Match All</strong></td>
</tr>
</tbody>
</table>
| Header List    | Specifies individual headers and their values that the BIG-IP system uses as criteria for routing HTTP requests. Using the **Entry Type** list, you must also identify each header as either a pattern string or a regular expression. This setting appears only when the value of **Headers** is **Match Only**.  
*Note:* When you use pattern strings, this list type is case-sensitive. For more information, see Configuring an HTTP Class profile, on page 8-11. | No default value |
| Cookies        | Specifies whether cookies used as criteria for routing those requests constitute all cookies or individual cookies that you specify. A value of **Match All** directs the system to forward HTTP requests based on all cookies. A value of **Match Only** directs the system to forward HTTP requests based on only those cookies you specify. | **Match All** |
| Cookie List    | Specifies individual cookies to be used as criteria for routing HTTP requests. Using the **Entry Type** list, you must also identify each cookie as either a pattern string or a regular expression. This setting appears only when the value of **Cookies** is **Match Only**.  
*Note:* When you use pattern strings, this list type is case-sensitive. For more information, see Configuring an HTTP Class profile, on page 8-11. | No default value |
| Send To        | Specifies the destination for HTTP traffic. Possible values are **None**, **Pool**, or **Redirect To**.                                                                                                       | **None**      |
| Pool           | Specifies the name of the pool to which you want to send classified traffic. This setting appears only when the value of the **Send To** setting is **Pool**.                                                          | **None**      |
Classifying traffic through other BIG-IP modules

If the BIG-IP system includes the Application Security Manager or WebAccelerator system, you can configure the system to send HTTP traffic to that module before sending the traffic to its final destination. For example, you can use an HTTP Class profile to instruct a virtual server to send traffic through Application Security Manager before forwarding the traffic to a load balancing pool.

You can create an HTTP Class profile from the Local Traffic section of the Configuration utility or from within the Application Security Manager or WebAccelerator system. Note that when you classify traffic from within Application Security Manager, the HTTP Class profile is known as an application security class.

If you create the profile (or application security class) from within a module, the module is already enabled by default. Conversely, if you create an HTTP Class profile using the Local Traffic section of the Configuration utility, you must explicitly enable the Application Security or WebAccelerator setting from within the profile. If you do not explicitly enable this setting, you effectively disable the module for the associated virtual server.

---

### Table 8.4 Settings of an HTTP Class profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redirect To Location</td>
<td>Specifies the URI to which the system should send the traffic. You use this setting when you want the profile to redirect the client request from an HTTP virtual server to an HTTPS virtual server, instead of to a pool. For example, you can create an HTTP virtual server with the URL <a href="http://siterequest/">http://siterequest/</a>, to listen on port 80. You can then assign an HTTP Class profile to the virtual server, to redirect client requests to the HTTPS virtual server, <a href="https://siterequest/">https://siterequest/</a>. Note that the string you specify can be a Tcl expression, such as https://[HTTP::host][HTTP::uri].</td>
<td>No default value</td>
</tr>
<tr>
<td>Rewrite URI</td>
<td>Specifies the Tcl expression that the system uses to rewrite the request URI that is forwarded to the server without sending an HTTP redirect to the client. Note that if you use static text for this setting instead of a Tcl expression, the system maps the specified URI for every incoming request. Also, you cannot use this setting if the value of the Send To setting is Redirect To.</td>
<td>No default value</td>
</tr>
</tbody>
</table>
For more information on configuring Application Security Manager application security classes and WebAccelerator HTTP Class profiles, see these documents:

- *Configuration Guide for BIG-IP® Application Security Management*
- *Configuration Guide for the BIG-IP® WebAccelerator™ System*
Configuring TCP profiles

TCP profiles are configuration tools that help you to manage TCP network traffic. Many of the configuration settings of TCP profiles are standard SYSCTL types of settings, while others are unique to the BIG-IP system. TCP profiles are important because they are required for implementing certain types of other profiles. For example, by implementing TCP, HTTP, and OneConnect profiles, along with a persistence profile and a remote authentication profile, you can take advantage of these traffic management features:

- Content spooling, to reduce server load
- OneConnect, to pool server-side connections
- Layer 7 session persistence, such as hash or cookie persistence
- iRules for managing HTTP traffic
- HTTP RAM Cache
- HTTP data compression
- HTTP pipelining
- Application authentication using a remote server
- Rewriting of HTTP redirections

The BIG-IP system contains three specific TCP profiles: the default TCP profile (tcp), and two custom TCP profiles (tcp-lan-optimized, and tcp-wan-optimized) that F5 Networks has created for you. You can implement any one of these profiles 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 8.5, on page 8-17 lists and describes the settings of a TCP profile.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>tcp</td>
</tr>
<tr>
<td>Reset on Timeout</td>
<td>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.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Time Wait Recycle</td>
<td>Recycles the connection when a SYN packet is received in a TIME-WAIT state.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Delayed ACKs</td>
<td>If this setting is enabled, allows coalescing of multiple acknowledgement (ACK) responses.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Proxy Maximum Segment</td>
<td>Advertises the same maximum segment to the server as was negotiated with the client.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Proxy Options</td>
<td>Advertises an option (such as timestamps) to the server only if it was negotiated with the client.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Proxy Buffer Low</td>
<td>Specifies the proxy buffer level at which the receive window was opened.</td>
<td>4096</td>
</tr>
<tr>
<td>Proxy Buffer High</td>
<td>Specifies the proxy buffer level at which the receive window was closed.</td>
<td>16384</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the number of seconds that a connection is idle before the connection is eligible for deletion. For background information on setting idle timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
<tr>
<td>Time Wait</td>
<td>Specifies the number of milliseconds that a connection is in a TIME-WAIT state before entering the CLOSED state.</td>
<td>2000</td>
</tr>
<tr>
<td>FIN Wait</td>
<td>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).</td>
<td>5</td>
</tr>
<tr>
<td>Close Wait</td>
<td>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).</td>
<td>5</td>
</tr>
<tr>
<td>Send Buffer</td>
<td>Causes the BIG-IP system to send the buffer size, which is specified in bytes.</td>
<td>32768</td>
</tr>
<tr>
<td>Receive Window</td>
<td>Causes the BIG-IP system to receive the window size, which is specified in bytes.</td>
<td>32768</td>
</tr>
</tbody>
</table>

*Table 8.5 Settings of a TCP profile*
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep Alive Interval</td>
<td>Causes the BIG-IP system to keep alive the probe interval, which is specified in seconds.</td>
<td>1800</td>
</tr>
<tr>
<td>Maximum SYN Retransmissions</td>
<td>Specifies the maximum number of retransmissions of SYN segments that the BIG-IP system allows.</td>
<td>3</td>
</tr>
<tr>
<td>Maximum Segment Retransmissions</td>
<td>Specifies the maximum number of retransmissions of data segments that the BIG-IP system allows.</td>
<td>8</td>
</tr>
<tr>
<td>IP ToS</td>
<td>Specifies the Type of Service level that the BIG-IP system assigns to TCP packets when sending them to clients.</td>
<td>0</td>
</tr>
<tr>
<td>Link QoS</td>
<td>Specifies the Quality of Service level that the BIG-IP system assigns to TCP packets when sending them to clients.</td>
<td>0</td>
</tr>
<tr>
<td>Selective ACKs</td>
<td>Specifies, when checked (enabled), that the system processes data using selective ACKs whenever possible, to improve system performance. Enabling this setting improves packet flow in a lossy network because the system can acknowledge successfully received packets out of order. This is a negotiated option and is automatically disabled if not supported by a peer. Note: F5 recommends that you use the default value.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Extended Congestion Notification</td>
<td>Specifies, when checked (enabled), that the system uses the TCP flags <strong>CWR</strong> (congestion window reduction) and <strong>ECE</strong> (ECN-Echo) to notify its peer of congestion and congestion counter-measures. Note: F5 recommends that you use the default setting. When enabled, this setting can interfere with overall congestion calculations. The setting also allows for potential security issues, whereby an intermediate device can stimulate poor performance by spoofing CWR packets.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Extensions for High Performance (RFC 1323)</td>
<td>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. These options are used to help calculate the round trip time, as well as the available resources on a peer. They are fundamentally linked with congestion control. Also, these options are normally negotiated, and you should not need to disable them unless a network device or peer does not implement them correctly.</td>
<td>Enabled (Checked)</td>
</tr>
</tbody>
</table>

*Table 8.5 Settings of a TCP profile*
Limited Transmit Recovery

- **Description**: 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. Enabling this setting allows TCP to temporarily stretch the congestion window when first receiving a duplicate ACK packet. This in turn allows for faster retransmissions and a quicker recovery from the small congestion window. With this setting enabled, the aggressive transmit behavior is limited to the recovery period.

- **Default Value**: Enabled (Checked)

Slow Start

- **Description**: Specifies, when checked (enabled), that the system uses larger initial window sizes (as specified in RFC 3390) to help reduce round trip times. The setting ramps up the amount of data transmitted to a peer over a period of time. Enabling this setting avoids sudden and excessive congestion on the link. Also, the congestion metrics cache might provide historical data about the peer, allowing the slow start to be jump started.

  - If you disable this setting, the system initializes the congestion window to the maximum window scale and attempts to transmit as much data as possible until congestion occurs. Consequently, in networks with unlimited bandwidth (such as directly-connected local peers), more data can initially be transmitted.

- **Default Value**: Enabled (Checked)

Deferred Accept

- **Description**: 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.

- **Default Value**: Disabled (Cleared)

Verified Accept

- **Description**: When enabled, verifies that a server is available to accept the connection (by actually sending the server a SYN) before responding to the client's SYN with a SYN-ACK. (Normally, the BIG-IP system accepts the client's connection before selecting a server with which to communicate.)

- **Default Value**: Disabled (Cleared)

Bandwidth Delay

- **Description**: Specifies, when checked (enabled), that the system attempts to calculate the optimal bandwidth to use to the client, based on throughput and round-trip time, without exceeding the available bandwidth.

- **Default Value**: Enabled (Checked)

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limited Transmit Recovery</td>
<td>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. Enabling this setting allows TCP to temporarily stretch the congestion window when first receiving a duplicate ACK packet. This in turn allows for faster retransmissions and a quicker recovery from the small congestion window. With this setting enabled, the aggressive transmit behavior is limited to the recovery period.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Slow Start</td>
<td>Specifies, when checked (enabled), that the system uses larger initial window sizes (as specified in RFC 3390) to help reduce round trip times. The setting ramps up the amount of data transmitted to a peer over a period of time. Enabling this setting avoids sudden and excessive congestion on the link. Also, the congestion metrics cache might provide historical data about the peer, allowing the slow start to be jump started. If you disable this setting, the system initializes the congestion window to the maximum window scale and attempts to transmit as much data as possible until congestion occurs. Consequently, in networks with unlimited bandwidth (such as directly-connected local peers), more data can initially be transmitted.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Deferred Accept</td>
<td>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.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Verified Accept</td>
<td>When enabled, verifies that a server is available to accept the connection (by actually sending the server a SYN) before responding to the client's SYN with a SYN-ACK. (Normally, the BIG-IP system accepts the client's connection before selecting a server with which to communicate.)</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Bandwidth Delay</td>
<td>Specifies, when checked (enabled), that the system attempts to calculate the optimal bandwidth to use to the client, based on throughput and round-trip time, without exceeding the available bandwidth.</td>
<td>Enabled (Checked)</td>
</tr>
</tbody>
</table>

*Table 8.5 Settings of a TCP profile*
# Nagle’s Algorithm

Specifies, when checked (enabled), that the system applies Nagle’s algorithm to reduce the number of short segments on the network. When the system receives packets that are less than the maximum segment size (MSS), the packets are coalesced until the peer has sent the ACK packet for the previous segment. This helps to reduce congestion by creating fewer packets on the network.

Note that enabling this setting for interactive protocols such as Telnet may cause degradation on high-latency networks.

Table 8.5 Settings of a TCP profile

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nagle’s Algorithm</td>
<td>Specifies, when checked (enabled), that the system applies Nagle’s algorithm to reduce the number of short segments on the network. When the system receives packets that are less than the maximum segment size (MSS), the packets are coalesced until the peer has sent the ACK packet for the previous segment. This helps to reduce congestion by creating fewer packets on the network. Note that enabling this setting for interactive protocols such as Telnet may cause degradation on high-latency networks.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Acknowledge on Push</td>
<td>Specifies, when enabled, significantly improved performance to Windows® and MacOS peers who are writing out on a very small send buffer.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>MD5 Signature</td>
<td>Specifies, when enabled, to use RFC2385 TCP-MD5 signatures to protect TCP traffic against intermediate tampering.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>MD5 Signature Passphrase</td>
<td>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.</td>
<td>No default value</td>
</tr>
<tr>
<td>Congestion Control</td>
<td>Specifies the congestion control mechanism that the BIG-IP system is to use. Possible values are:</td>
<td>New Reno</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong>—No congestion control algorithm implemented.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>High Speed</strong>—A more aggressive, loss-based algorithm.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>New Reno</strong>—A modification to the Reno algorithm, that responds to partial acknowledgements when selective acknowledgements (SACK) are unavailable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Reno</strong>—An implementation of the TCP Fast Recovery algorithm, based on the implementation in the BSD Reno release.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Scalable</strong>—A TCP algorithm modification that adds a scalable, delay-based and loss-based component into the Reno algorithm.</td>
<td></td>
</tr>
<tr>
<td>Congestion Metrics Cache</td>
<td>Specifies, when checked (enabled), that the system uses a cache for storing congestion metrics. Subsequently, because these metrics are already known and cached, the initial slow-start ramp for previously-encountered peers improves.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Appropriate Byte Counting</td>
<td>Increases the congestion window by basing the increase amount on the number of previously unacknowledged bytes that each ACK covers.</td>
<td>Enabled (Checked)</td>
</tr>
</tbody>
</table>

Note: F5 recommends that you use the default setting. When this setting is disabled, in situations with lost ACK packets, the congestion window remains small for a longer period of time.
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 the 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.*

### Understanding tcp-lan-optimized profile settings

The tcp-lan-optimized profile is a TCP-type profile. This profile is effectively a custom profile that the BIG-IP system has already created for you, derived from the default tcp profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-SACK (RFC 2883)</td>
<td>Specifies the use of the Selective ACKs (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.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Packet Lost Ignore Rate</td>
<td>Specifies the threshold of packets lost per million at which the system performs congestion control. Valid values range from 0 to 1,000,000. The default is 0, meaning the system performs congestion control if any packet loss occurs. If you set the ignore rate to 10 and packet loss for a TCP connection is greater than 10 per million, congestion control occurs.</td>
<td>0</td>
</tr>
<tr>
<td>Packet Lost Ignore Burst</td>
<td>Specifies the probability of performing congestion control when multiple packets are lost, even if the value of the Packet Lost Ignore Rate setting was not exceeded. Valid values range from 0 to 4,294,967,295. A value of 0 means that the system performs congestion control if any packets are lost. Higher values decrease the chance of performing congestion control.</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 8.5 Settings of a TCP profile*
The **tcp-lan-optimized** profile inherits its settings and their default values from the **tcp** profile, but some of the setting values have been changed. By implementing the **tcp-lan-optimized** profile, you can optimize the performance of your local TCP traffic in certain ways, without having to create a custom profile to do so.

You can use the **tcp-lan-optimized** profile as is, or you can create another custom profile, specifying the **tcp-lan-optimized** profile as the parent profile.

The default setting values of the **tcp-lan-optimized** profile are the same as those of the **tcp** profile, except for those listed in Table 8.6.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Buffer Low</td>
<td>Specifies the proxy buffer level at which the receive window was opened.</td>
<td>98304</td>
</tr>
<tr>
<td>Proxy Buffer High</td>
<td>Specifies the proxy buffer level at which the receive window was closed.</td>
<td>131072</td>
</tr>
<tr>
<td>Send Buffer</td>
<td>This setting causes the BIG-IP system to send the buffer size, in bytes.</td>
<td>65535</td>
</tr>
<tr>
<td>Receive Window</td>
<td>This setting causes the BIG-IP system to receive the window size, in bytes.</td>
<td>65535</td>
</tr>
<tr>
<td>Slow Start</td>
<td>This setting specifies, when checked (enabled), that the system uses larger initial window sizes (as specified in RFC 3390) to help reduce round trip times.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Bandwidth Delay</td>
<td>This setting specifies, when checked (enabled), that the system attempts to calculate the optimal bandwidth to use to the client, based on throughput and round-trip time, without exceeding the available bandwidth.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Nagle's Algorithm</td>
<td>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.</td>
<td>Disabled (Cleared)</td>
</tr>
<tr>
<td>Acknowledge on Push</td>
<td>Specifies, when enabled, significantly improved performance to Windows and MacOS peers who are writing out on a very small send buffer.</td>
<td>Enabled (Checked)</td>
</tr>
</tbody>
</table>

Table 8.6  Values of a **tcp-lan-optimized** profile that are different from the **tcp** profile

### Understanding tcp-wan-optimized profile settings

The **tcp-wan-optimized** profile is a TCP-type profile. This profile is effectively a custom profile that the BIG-IP system has already created for you, derived from the default **tcp** profile.
The `tcp-wan-optimized` profile inherits its settings and their default values from the `tcp` profile, but some of the setting values have been changed. By implementing the `tcp-wan-optimized` profile, you can optimize the performance of your wide-area TCP traffic in certain ways, without having to create a custom profile to do so.

You can use the `tcp-wan-optimized` profile as is, or you can create another custom profile, specifying the `tcp-wan-optimized` profile as the parent profile.

The default setting values of the `tcp-wan-optimized` profile are the same as those of the `tcp` profile, except for those listed in Table 8.7.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Buffer Low</td>
<td>Specifies the proxy buffer level at which the receive window was opened.</td>
<td>131072</td>
</tr>
<tr>
<td>Proxy Buffer High</td>
<td>Specifies the proxy buffer level at which the receive window was closed.</td>
<td>131072</td>
</tr>
<tr>
<td>Send Buffer</td>
<td>This setting causes the BIG-IP system to send the buffer size, in bytes.</td>
<td>65535</td>
</tr>
<tr>
<td>Receive Window</td>
<td>This setting causes the BIG-IP system to receive the window size, in bytes.</td>
<td>65535</td>
</tr>
<tr>
<td>Selective ACKs</td>
<td>This setting specifies, when checked (enabled), that the system processes data using selective ACKs whenever possible, to improve system performance.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Nagle's Algorithm</td>
<td>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.</td>
<td>Enabled (Checked)</td>
</tr>
</tbody>
</table>

*Table 8.7 Values of a *tcp-wan-optimized* profile that are different from the *tcp* profile*
Configuring a UDP profile

The UDP profile is a configuration tool for managing UDP network traffic. Table 8.8 lists and describes the settings of a UDP profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This setting specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>udp</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>This setting specifies the number of seconds that a connection is idle before the connection flow is eligible for deletion. For background information on setting idle timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>60</td>
</tr>
<tr>
<td>IP ToS</td>
<td>This setting specifies the Type of Service level that the BIG-IP system assigns to UDP packets when sending them to clients.</td>
<td>0</td>
</tr>
<tr>
<td>Link QoS</td>
<td>This setting specifies the Quality of Service level that the BIG-IP system assigns to UDP packets when sending them to clients.</td>
<td>0</td>
</tr>
<tr>
<td>Datagram LB</td>
<td>This setting specifies, when checked (enabled), that the system load balances UDP traffic packet-by-packet.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Allow No Payload</td>
<td>This setting specifies, when checked (enabled), that the system passes datagrams that contain header information, but no essential data.</td>
<td>Disabled (Unchecked)</td>
</tr>
</tbody>
</table>

*Table 8.8 Settings of a UDP profile*
Configuring an SCTP profile

The BIG-IP system includes a profile type that you can use to manage Stream Control Transmission Protocol (SCTP) traffic. Stream Control Transmission Protocol (SCTP) is a general-purpose, industry-standard transport protocol, designed for message-oriented applications that transport signalling data. The design of SCTP includes appropriate congestion avoidance behavior, as well as resistance to flooding and masquerade attacks.

Unlike TCP, SCTP includes the ability to support several streams within a connection. While a TCP stream refers to a sequence of bytes, an SCTP stream represents a sequence of messages.

You can use SCTP as the transport protocol for applications that require monitoring and detection of session loss. For such applications, the SCTP mechanisms to detect session failure actively monitor the connectivity of a session.

You can tailor SCTP 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 5, Understanding Profiles.

You can use the default sctp profile as is, or create a custom SCTP profile. Table 8.9 lists and describes the settings of an SCTP profile.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>tcp</td>
</tr>
<tr>
<td>Receive Ordered</td>
<td>If this setting is enabled, the system delivers messages to an upper layer, in order.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Send Partial</td>
<td>If this setting is enabled, the system accepts a partial amount of application data.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>TCP Shutdown</td>
<td>If this setting is enabled, SCTP instances emulate TCP closing. After receiving a SHUTDOWN message from an upper-layer user process, an SCTP instance initiates a graceful shutdown, by sending a SHUTDOWN chunk.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Reset on Timeout</td>
<td>If this setting is enabled and an SCTP connection exceeds the timeout value for idle connections, the system sends a reset in addition to deleting the connection.</td>
<td>Enabled (Checked)</td>
</tr>
</tbody>
</table>

Table 8.9 Settings of an SCTP profile
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out Streams</td>
<td>Specifies the number of outbound streams that you want the chunk to request.</td>
<td>2</td>
</tr>
<tr>
<td>In Streams</td>
<td>Specifies the number of inbound streams that you want the chunk to request.</td>
<td>2</td>
</tr>
<tr>
<td>Send Buffer</td>
<td>Causes the BIG-IP system to send the buffer size, in bytes.</td>
<td>65536</td>
</tr>
<tr>
<td>Receive Window</td>
<td>Specifies the number of bytes that a sender can transmit without receiving an acknowledgment (ACK).</td>
<td>65535</td>
</tr>
<tr>
<td>Transmit Chunks</td>
<td>Specifies the number of transmit chunks allowed in buffer.</td>
<td>256</td>
</tr>
<tr>
<td>Receive Chunks</td>
<td>Specifies the number of receive chunks allowed in buffer.</td>
<td>256</td>
</tr>
<tr>
<td>Cookie Expiration</td>
<td>Specifies the valid duration of a cookie, in seconds.</td>
<td>60</td>
</tr>
<tr>
<td>Maximum Initial Retransmit Limit</td>
<td>Specifies the maximum number of times that the system attempts to establish a connection.</td>
<td>4</td>
</tr>
<tr>
<td>Maximum Association Retransmit Limit</td>
<td>Specifies the maximum number of times that the system attempts to send data.</td>
<td>8</td>
</tr>
<tr>
<td>Proxy Buffer Low</td>
<td>Specifies the proxy buffer level at which the system opens the receive window.</td>
<td>4096</td>
</tr>
<tr>
<td>Proxy Buffer High</td>
<td>Specifies the proxy buffer level at which the system closes the receive window.</td>
<td>16384</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the number of seconds that a connection is idle before the connection is eligible for deletion. For background information on setting idle timeout values, see Chapter 1, <em>Introducing Local Traffic Management</em>.</td>
<td>300</td>
</tr>
<tr>
<td>Heartbeat Interval</td>
<td>Specifies the number of seconds to wait before sending a heartbeat chunk.</td>
<td>30</td>
</tr>
<tr>
<td>IP ToS to Peer</td>
<td>Specifies the Type of Service level that the BIG-IP system assigns to SCTP packets when sending them to a client.</td>
<td>0</td>
</tr>
<tr>
<td>Link QoS to Peer</td>
<td>Specifies the Quality of Service level that the BIG-IP system assigns to SCTP packets when sending them to a client.</td>
<td>0</td>
</tr>
<tr>
<td>Secret</td>
<td>Specifies the internal secret string used to calculate the key-hash method authentication code (HMAC) for cookie verification.</td>
<td>default</td>
</tr>
</tbody>
</table>

*Table 8.9 Settings of an SCTP profile*
Managing SSL Traffic

• Introducing SSL traffic management
• Managing keys and certificates
• Understanding SSL profiles
• Configuring SSL profile settings
• Configuring client and server authentication settings
Introducing SSL traffic management

The BIG-IP® local traffic management system offers several features that you can use to intelligently control your SSL traffic. Some of the SSL traffic-management features are:

- The ability to authenticate clients and servers to maintain secure connections between a client system and the BIG-IP system, and between the BIG-IP system and a target web server.
- The ability to offload SSL certificate-verification tasks from client and server systems.
- iRule commands that provide features such as header insertion and HTTP redirection.
- Support for Online Certificate Status Protocol (OCSP), for the purpose of obtaining up-to-date certificate revocation status.

The primary way that you can control SSL network traffic is by configuring a Client or Server SSL profile. A **Client profile** is a type of traffic profile that enables the BIG-IP system to accept and terminate any client requests that are sent by way of a fully SSL-encapsulated protocol. A **Server profile** is a type of profile that enables the BIG-IP system to initiate secure connections to a target web server.

Managing SSL traffic requires you to complete these tasks:

- Install a key/certificate pair on the BIG-IP system for terminating client-side secure connections.
- Configure a profile, and then associate that profile with a virtual server. You configure a profile using the Configuration utility.
- Associate the profile with a virtual server.

As an option, you can write an iRule that gives you the ability to manage individual SSL connections in certain ways. For more information, see Chapter 17, *Writing iRules*.

Managing client-side and server-side traffic

With the BIG-IP system, you can enable SSL traffic management for either client-side traffic or server-side traffic.

**Client-side traffic** refers to connections between a client system and the BIG-IP system. **Server-side traffic** refers to connections between the BIG-IP system and a target server system.
Managing client-side SSL traffic
When you enable the BIG-IP system to manage client-side SSL traffic, the BIG-IP system terminates incoming SSL connections by decrypting the client request. The BIG-IP system then sends the request, in clear text, to a target server. Next, the BIG-IP system retrieves a clear-text response (such as a web page) and encrypts the request, before sending the web page back to the client. During the process of terminating an SSL connection, the BIG-IP system can, as an option, perform all of the SSL certificate verification functions normally handled by the target web server. For information on configuring a client-side SSL profile, see Understanding SSL profiles, on page 9-11.

Managing server-side SSL traffic
When you enable the BIG-IP system to manage server-side SSL traffic, the BIG-IP system enhances the security of your network by re-encrypting a decrypted request before sending it on to a target server. In addition to this re-encryption, the BIG-IP system can, as an option, perform the same verification functions for server certificates that the BIG-IP system can for client certificates. For information on configuring a server-side SSL profile, see Understanding SSL profiles, on page 9-11.

Summarizing SSL traffic-control features
The BIG-IP system includes several important features related to controlling SSL traffic. Foremost among these features are SSL authentication (that is, certificate verification and revocation), as well as encryption and decryption. While most of these features are available through configuration of a Client or Server SSL profile, others are offered through features such as iRules™. Table 9.1 summarizes the features related to SSL traffic management; the sections following the table describe these features.

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
<th>Configuration Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate verification</td>
<td>In terminating and initiating SSL connections, the BIG-IP system can perform a full range of certificate verification tasks, including the verification of both client and server certificates. For example, you can define the extent to which the system verifies client certificates: you can configure the system to request client certificates, require them, or ignore them altogether. You can also specify settings such as the depth of certificate chain traversal.</td>
<td>Client and Server SSL profiles iRules</td>
</tr>
<tr>
<td>Certificate revocation</td>
<td>When a client or server presents a certificate to the BIG-IP system, the system can check the revocation status of a certificate as part of the certificate verification process.</td>
<td>Client and Server SSL profiles OCSP profile</td>
</tr>
</tbody>
</table>

Table 9.1 Summary of the BIG-IP system features related to SSL traffic control
Managing SSL Traffic

Understanding certificate verification

Certificate verification is the process of determining whether a client or server can trust a certificate that is presented by a peer (that is, a client or a server). When receiving a request from a client or a response from a server, the BIG-IP system attempts to verify that the certificate presented by the client or server can be trusted.

Signed certificates

A basic security requirement for clients and servers handling SSL connections is that they each present a certificate signed by a trusted Certificate Authority (CA), whenever they communicate with a peer. As an option, you can configure the BIG-IP system to handle the task of verifying client certificates.

When you configure certificate verification on the BIG-IP system, the BIG-IP system must hold a certificate that it can present to clients when processing SSL requests.

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
<th>Configuration Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption and decryption</td>
<td>As a way to off-load work from target web servers, the BIG-IP system decrypts incoming client requests. As an added security option, you can configure the profile to re-encrypt a request before forwarding it on to a server. Encryption and decryption of requests and responses are based on particular ciphers that you specify as part of SSL profile configuration.</td>
<td>Client and Server SSL profiles</td>
</tr>
<tr>
<td>Authorization</td>
<td>You can control access to system resources by configuring the BIG-IP system in certain ways. For example, you can create an iRule to insert client certificate information into client requests and then grant access based on that information. Also, for environments that include an LDAP database server, the BIG-IP system can grant access to resources by querying the LDAP server using client certificate credentials. You can also limit the number of concurrent TCP connections.</td>
<td>SSL Client Certificate LDAP profile iRules</td>
</tr>
<tr>
<td>SSL session persistence</td>
<td>A powerful feature of the BIG-IP system is its ability to enable persistence for SSL sessions. Two types of SSL persistence are available, depending on whether you have configured the BIG-IP system to terminate SSL connections.</td>
<td>Client and Server SSL profiles SSL Persistence profile iRules</td>
</tr>
<tr>
<td>Other SSL features</td>
<td>In addition to the features listed above, the BIG-IP system allows you to configure other options such as invalid protocol versions, the size and timeout values of the SSL session cache, and SSL shutdown behavior.</td>
<td>Client and Server SSL profiles</td>
</tr>
</tbody>
</table>

Table 9.1  Summary of the BIG-IP system features related to SSL traffic control
Optionally, you can configure the BIG-IP system to verify server responses. In this case, if the server specifically requests a certificate from the BIG-IP system, the BIG-IP system needs to hold a second certificate.

If you configure the BIG-IP system to verify client and server certificates, you must generate and install keys and certificates onto the BIG-IP system before the system can manage SSL traffic. You can do this through the Configuration utility. As an alternative to generating new key/certificate pairs, you can import existing ones. For more information on importing existing key/certificate pairs, see *Importing keys, certificates, and archives*, on page 9-10.

**Client-side certificate verification**

When a client makes an HTTP request, the BIG-IP system can perform the client certificate verification task that is normally performed by the target server.

When a client presents a certificate to the BIG-IP system, the BIG-IP system uses a *client trusted CAs file* to determine the Certificate Authorities that it can trust. Using this file is the primary way that the BIG-IP system attempts to verify a client certificate. When you create a client-side profile, the BIG-IP system automatically creates a default client trusted CAs file. You can either use the default file name specified in the profile, or you can specify a different file name. For more information, see *Specifying trusted client CAs*, on page 9-16.

A second factor in client certificate verification is the *client certificate CA file*, which the BIG-IP system uses to advertise to clients a list of the CAs that the profile trusts. Not that this list could be different from the list of CAs that the BIG-IP system actually trusts. As with the trusted CAs file, the BIG-IP system automatically creates a a default version of this file.

There is also a client chain file, which the BIG-IP system sends to a client as part of the client certificate verification process. The default client chain file is the Client Trusted CAs file. For more information, see *Specifying trusted client CAs*, on page 9-16.

**Server-side certificate verification**

Server-side verification occurs when you enable a Server SSL profile and set the presentation of a server certificate to *require*.

When a server presents a certificate to the BIG-IP system, the BIG-IP system uses the *server trusted CAs file* to determine which Certificate Authorities it can trust. Using this file is the primary way that the BIG-IP system attempts to verify a server certificate. The BIG-IP system automatically creates a default Server Trusted CAs file when you configure a server-side profile. You can either use the default file name specified in the profile, or specify a different file name.

There is also a server chain file, which the BIG-IP system sends to a server as part of the entire server certificate verification process. The default server chain file is the Server Trusted CAs file.
Understanding certificate revocation

The BIG-IP system can check to see if a certificate being presented by a client or server has been revoked. A revoked client certificate indicates to the BIG-IP system that the system should fail to authenticate the client.

The BIG-IP system supports two industry-standard methods for checking the revocation status of a certificate. These two methods are:

- **Certificate revocation lists (CRLs).**
  CRLs are a method that the BIG-IP system can use to check whether a certificate being presented to the BIG-IP system has been revoked. This CRL support is in the form of a CRL file and a CRL path. The BIG-IP system enables you to configure one CRL file and path for the client-side profile, and one CRL file and path for the server-side profile. You configure the use of CRLs through an SSL profile. For more information, see Configuring client and server authentication settings, on page 9-25.

- **Online Certificate Status Protocol (OCSP)**
  Unlike the use of CRLs, OCSP ensures that the revocation status of a certificate is always up-to-date. You configure OCSP through an Authentication profile. For more information, see Chapter 10, Authenticating Application Traffic.

Understanding encryption/decryption

Another feature of the BIG-IP system is its ability to handle encryption and decryption tasks that a target server normally performs as part of processing a client request. Using a client-side SSL profile, the BIG-IP system decrypts incoming requests before sending them on in plain text to the target server. Using a server-side profile, the BIG-IP system provides an additional level of security by re-encrypting the request before sending it on to the target server.

A BIG-IP system feature related to encryption is the ability to insert into incoming HTTP requests the cipher that the client used to encrypt its request. You can then direct the traffic based on that cipher specification. For more information on specifying ciphers to control the destination of client requests, see Chapter 17, Writing iRules.
Chapter 9

Understanding client authorization

Unlike authentication, *authorization* is not about trusting identity, but about controlling access to system resources. Once the SSL profile has verified that a client or server can be trusted, the BIG-IP system can then control the connection’s level and type of access to the destination content.

The BIG-IP system features that support access control for SSL connections are:

- Inserting client certificate fields into HTTP requests
- Limiting the number of concurrent client TCP connections
- Client authorization with an LDAP database server

One of the most useful ways to control a client’s access to system resources is to create an iRule that inserts fields of a client certificate as headers into client requests. For example, an iRule can insert the status of a client certificate as a header into a request, and then use the `HTTP::header` command to select the target server based on that status.

For more information on using this header insertion feature to control the destination of client requests, see Chapter 17, *Writing iRules*.

Understanding SSL session persistence

There are two types of SSL persistence available that you can implement. The first type is the standard SSL persistence mode, which enables persistence for SSL sessions that do not involve the decryption of SSL requests and the re-encryption of SSL responses. You enable this SSL persistence mode by configuring an SSL persistence profile. For more information, see Chapter 7, *Enabling Session Persistence*.

The second type of SSL persistence enables persistence for SSL sessions that involve the decryption of requests and re-encryption of responses. In this case, you implement SSL persistence by inserting SSL session IDs as headers into HTTP requests. You insert session ID headers by writing an iRule. For information on iRules, see Chapter 17, *Writing iRules*.

Understanding other SSL features

In addition to using the preceding features, you can also perform tasks such as specifying invalid protocol versions, configuring the size and timeout values of the SSL session cache, and configuring SSL shutdown behavior. You can also configure an SSL profile to renegotiate SSL sessions based on timeout values and amount of data transmitted.
Managing keys and certificates

Before you can enable decryption and encryption of SSL connections, you must install one or more SSL certificates on the BIG-IP system. The purpose of these certificates is described in the section *Understanding certificate verification*, on page 9-3.

To ease the task of generating certificate requests and sending them to certificate authorities, the BIG-IP system provides a set of key management screens within the Configuration utility. You access these certificate management screens from the *Local Traffic* section of the Main tab.

When managing keys, you can:

- Display information about all existing key pairs and certificates.
- Create requests for new key pairs and certificates and submit those requests to certificate authorities.
- Renew certificate requests.
- Display key and certificate properties.
- Import and export PEM-formatted keys and certificates.

Displaying information about existing keys and certificates

Summary information about existing key pairs and certificates is available from the Configuration utility. The SSL Certificates screen displays the following information:

- Status of the certificate (valid, about to expire, or expired)
- The unique name that you assigned the certificate when creating the request
- The issuer of the certificate
- The expiration date

To display information about existing certificates and keys

1. On the Main tab of the navigation pane, expand *Local Traffic*, and click **SSL Certificates**. This opens the SSL Certificates screen and lists all certificates installed on the BIG-IP system.
2. In the Name column, click a certificate name. This displays the properties of that certificate.
3. If you want to see information about the key that is associated with that certificate, click **Key** on the menu bar. This displays the type and size of the key.
Creating a request for a new certificate and key

Using the Configuration utility, you can either generate a self-signed certificate (usually used for internal test purposes only) or you can create a request for a certificate/key pair, to be sent to a certificate authority. When you send a request to a certificate authority, the certificate authority returns a signed certificate.

You can send a certificate request to a certificate authority in either of two ways:

• You can copy the text of the newly-generated request from the Configuration utility screen and give it to the certificate authority (using cut and paste).
• You can download the newly-generated request to a file and transmit the file to the certificate authority.

The way to transmit the request to a certificate authority (either through pasting the text or through a file attachment) is by accessing the certificate authority’s web site. The Configuration utility screen for submitting a request for signature by a certificate authority includes links to various certificate authority web sites.

To request a self-signed certificate

1. On the Main tab of the navigation pane, expand Local Traffic, and click SSL Certificates.
   This displays the SSL Certificates screen.
2. On the upper-right portion of the screen, click Create.
3. Type a unique name for the certificate.
4. For the Issuer setting, select Self.
5. Configure the Common Name setting and any other settings you want.
6. In the Key Properties section, select a key size.
7. Click Finished.

To request a certificate from a certificate authority

1. On the Main tab of the navigation pane, expand Local Traffic, and click SSL Certificates.
   This displays the SSL Certificates screen.
2. On the upper-right portion of the screen, click Create.
3. Type a unique name for the certificate.
4. In the Issuer box, select Certificate Authority.
5. Configure the Common Name setting and any other settings.
6. Click Finished.
   This displays your certificate request.
7. To download the request into a file on your system, either:
   • Copy the certificate from the Request Text box.
   • Click the button in the Request File box.

8. In the Certificate Authorities box, click a certificate authority name.
   This displays the web site for the certificate authority.

9. Follow the instructions on the web site for either pasting the copied request or attaching the generated request file.

10. Click Finished.

Renewing a certificate

All certificates include expiration dates. When a certificate expires, you must renew that certificate if you want to continue using it.

When renewing a self-signed certificate, you can explicitly set an expiration date by changing the value of the Lifetime setting. The default value of this setting, in days, is 365.

To renew a certificate

1. On the Main tab of the navigation pane, expand Local Traffic, and click SSL Certificates.
   This displays a list of existing certificates.

2. In the Name column, click the name of the certificate that you want to renew.
   This displays the certificate properties.

3. At the bottom of the screen, click Renew.

4. Modify or retain the settings.
   Note that a value for the Common Name setting is required.

5. Click Finished.

6. Exit the Configuration utility and access the bigpipe shell.

7. At the bigpipe shell prompt, type the load command.
   
   *Note: Typing the bigpipe utility load command is required to complete the certificate renewal operation.*
Deleting a certificate/key pair

You can delete certificates and keys that you no longer need.

**To delete a certificate/key pair**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SSL certificates**. 
   This displays the list of existing certificates.
2. Check the Select box to the left of the name of the certificate you want to delete.
3. At the bottom of the screen, click **Delete**. 
   A confirmation screen appears.
4. Click **Delete**. 
   This removes the certificate.

◆ **Note**

On the SSL certificates list screen, if you click a certificate name, display the properties of the certificate, and click the **Delete** button, you are only deleting the certificate portion of the certificate/key pair. Similarly, if you display the properties of the key and click the **Delete** button, you are only deleting the key portion of the pair. To fully delete the certificate, you must use the procedure shown above.

Importing keys, certificates, and archives

If you have transferred a key/certificate pair, a certificate, or a key/certificate archive onto the BIG-IP system from another system, and the certificate or archive is in the form of a file or a base-64 encoded text string, you can import this certificate or archive into the Configuration utility. By importing a certificate or archive into the Configuration utility, you ease the task of managing that certificate or archive. You can use the Import SSL Certificates and Keys screen only when the certificate you are importing is in Privacy Enhanced Mail (PEM) format.

**To import a key pair, certificate, or archive**

1. On the Main tab of the navigation pane, expand **Local Traffic**, click **SSL Certificates**. 
   This displays the list of existing certificates.
2. On the upper right corner of the screen, click **Import**.
3. From the **Import Type** list, select the type of import (**Key**, **Certificate**, or **Archive**). 
   This expands the configuration screen for that import type.
Managing SSL Traffic

4. Configure the settings required by your selection of import type:
   - **Key** or **Certificate**, configure the **Name** and **Source** settings.
   - **Archive**, configure the **Upload Archive File** setting.

5. Click **Import** (for keys and certificates) or **Load** (for archives).

Creating an archive

You can create an archive for one or more keys and certificates. You create an archive when you want to export multiple keys and certificates to another system.

**To create an archive**

1. On the Main tab of the navigation pane, expand **Local Traffic**, click **SSL Certificates**.
   The SSL Certificates screen opens.

2. At the bottom of the list, click **Archive**.

3. Type a name for your archive file.
   The file will have a `.tgz` extension.

4. In the **Key List** area:
   a) In the **Available Keys** box, select a key that you want to archive.
   b) Use the Move button (<<) to transfer the key name to the **Keys to Archive** box.

5. In the **Certificate List** area:
   a) In the **Available Certificates** box, select a certificate that you want to archive.
   b) Use the Move button (<<) to transfer the certificate name to the **Certificates to Archive** box.

6. Click the **Generate & Download Archive** button.

Understanding SSL profiles

As described in Chapter 5, *Understanding Profiles*, a **profile** is a group of settings with values that determine the way that the BIG-IP system manages application-specific network traffic. One type of traffic that a profile can manage is SSL traffic. The most basic functions of an SSL profile are to offload certificate validation and verification tasks, as well as encryption and decryption, from your target web servers.
The two types of SSL profiles are:

- **Client profiles**
  Client profiles allow the BIG-IP system to handle authentication and encryption tasks for any SSL connection coming into a BIG-IP system from a client system. You implement this type of profile by using the default `clientssl` profile, or by creating a custom profile based on the `clientssl` profile.

- **Server profiles**
  Server profiles allow the BIG-IP system to handle encryption tasks for any SSL connection being sent from a BIG-IP system to a target server. A server SSL profile is able to act as a client by presenting certificate credentials to a server when authentication of the BIG-IP system is required. You implement this type of profile by using the default `serverssl` profile, or creating a custom profile based on the `serverssl` profile.

---

**Important**

Prior to enabling SSL processing, you must first generate either a valid x509 certificate from a Trusted Certificate Authority or a temporary certificate, and install it on the BIG-IP system. For more information on certificates, see **Signed certificates**, on page 9-3. For instructions on how to generate and install a certificate on the BIG-IP system, see **Managing keys and certificates**, on page 9-7.

The settings that make up a Client or Server SSL profile are organized into three categories: General properties, Configuration, and either Client Authentication or Server Authentication. You can configure these settings at the time that you create an SSL profile or after profile creation by modifying the profile’s settings. For specific procedures on configuring a profile, see Chapter 5, **Understanding Profiles**.

---

**Tip**

For any individual SSL connection, you can override the value of an SSL profile setting. You do this by writing an iRule and including those SSL iRule commands that correspond to the settings you want to override. For more information, see Chapter 17, **Writing iRules**, and the F5 Networks DevCentral web site [http://devcentral.f5.com](http://devcentral.f5.com).
Configuring SSL profile settings

This section describes the settings that appear in the Configuration section of a Client SSL Profile or Server SSL Profile screen. For information on configuring the other SSL profile settings, see Configuring client and server authentication settings, on page 9-25.

For the procedure to create or modify an SSL profile, see Chapter 5, Understanding Profiles.

Table 9.2 shows the settings you can configure for a Client SSL or Server SSL profile. For those settings that have default values, you can retain those default settings or modify them. Following this table are descriptions of these settings.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the user-supplied name of the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the system-supplied profile from which a custom profile is derived.</td>
<td>clientssl or serverssl</td>
</tr>
<tr>
<td>Certificate</td>
<td>Specifies the name of the certificate installed on the BIG-IP system for the purpose of terminating or initiating an SSL connection.</td>
<td>default (Client) None (Server)</td>
</tr>
<tr>
<td>Key</td>
<td>Specifies the name of the key installed on the BIG-IP system for the purpose of terminating or initiating an SSL connection.</td>
<td>default (Client) None (Server)</td>
</tr>
<tr>
<td>Pass Phrase</td>
<td>Specifies the name of the pass phrase used to encrypt the key.</td>
<td>No default value</td>
</tr>
<tr>
<td>Confirm Pass Phrase</td>
<td>Confirms the name of the pass phrase used to encrypt the key.</td>
<td>No default value</td>
</tr>
<tr>
<td>Chain</td>
<td>Specifies or builds a certificate chain file that a client can use to authenticate the profile. For more information, see Configuring a certificate chain, on page 9-15.</td>
<td>None</td>
</tr>
<tr>
<td>Trusted Certificate Authorities</td>
<td>Configures certificate verification by specifying a list of client or server CAs that the BIG-IP system trusts. For more information, see Specifying trusted client CAs, on page 9-16.</td>
<td>None</td>
</tr>
<tr>
<td>Ciphers</td>
<td>Specifies the list of ciphers that the BIG-IP system supports. For more information, see Specifying SSL ciphers, on page 9-16.</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>Options</td>
<td>Specifies the value <strong>All Bugfixes Enabled</strong>, which enables a set of industry-related miscellaneous workarounds related to SSL processing. For more information, see Configuring workarounds, on page 9-17.</td>
<td>All Bugfixes Enabled</td>
</tr>
<tr>
<td>ModSSL Methods</td>
<td>Enables or disables ModSSL method emulation. This setting is disabled by default.</td>
<td>Disabled (unchecked)</td>
</tr>
</tbody>
</table>

Table 9.2 Configuration settings of an SSL profile
Before configuring an SSL profile, it is helpful to have a description of certain settings that you might want to change.

### Specifying a profile name

To create an SSL profile, you must specify a unique name for the profile. The **Name** setting is the only setting that you must actively specify when creating an SSL profile; all other settings have default values.

### Selecting a parent profile

Every profile that you create is derived from a parent profile. Using the **Parent Profile** setting, you can configure the default SSL profile as the parent profile, or you can configure another SSL profile that you have already created.
Specifying a certificate name

The value of the Certificate setting is the name of the certificate that you installed on the BIG-IP system for the purpose of authenticating client-side or server-side SSL connections. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, you can specify the name of an existing certificate, default. For more information on certificates, see Understanding certificate verification, on page 9-3.

Specifying a key name

The value of the Key setting is the name of the key that you installed on the BIG-IP system for the purpose of authenticating either client-side or server-side SSL connections. If you have not generated a key request nor installed a key on the BIG-IP system, you can specify the name of an existing key, default.

For more information on keys, see Understanding certificate verification, on page 9-3.

Specifying and confirming a pass phrase

With the Pass Phrase and Confirm Pass Phrase settings, you can specify a string to encrypt the SSL key. This feature is optional. You can access these settings by locating the Configuration list and selecting Advanced.

For added security, when you use these settings, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP system users, so there is no need to enable it.

Note that the length of an encrypted pass phrase exceeds the length of the unencrypted pass phrase. An example of an encrypted pass phrase is MDEyMzQ1Njc4OWFiY2RlZmdoaWprbG1ub3BxcnN0dXZ3eHl6.

Configuring a certificate chain

In any client verification process, not only does the BIG-IP system need to authenticate the client, but the client might need to authenticate the BIG-IP system. However, a certificate that the BIG-IP system uses to authenticate itself to a client is sometimes signed by an intermediate CA that is not trusted by that client. In this case, the BIG-IP system might need to use a certificate chain. The profile enables you to specify the name of a specific certificate chain file. Note that the certificate files that make up the chain file must be in PEM format.

When you configure the Chain setting, the possible values are None, ca-bundle, and default.
Specifying trusted client CAs

For client-side SSL processing, you can configure an SSL profile to verify certificates presented by a client or a server. Using the Trusted Certificate Authorities setting, you can specify a client trusted CAs file name, which the BIG-IP system then uses to verify client or server certificates. If you do not configure a trusted CAs file, the profile uses a default file.

The trusted CAs file that you specify for certificate verification contains one or more certificates, in Privacy Enhanced Mail (PEM) format. Built manually, this file contains a list of the client or server certificates that the SSL profile will trust. If you do not specify a trusted CAs file, or the specified trusted CAs file is not accessible to the BIG-IP system, the system uses the default file name.

When you configure the Trusted Certificate Authorities setting, the possible values are None, ca-bundle, and default.

Specifying SSL ciphers

For each SSL profile, you can specify the ciphers available for SSL connections. When configuring ciphers, you must ensure that the ciphers configured for the SSL profile match those of the client sending a request, or of the server sending a response.

For example, a client might connect to and successfully establish an SSL connection to an SSL profile that is configured to use both client-side and server-side SSL. After the client sends additional data (such as an HTTP request), the SSL profile attempts to establish an SSL connection to a server. However, the SSL profile might be configured to enable only 3DES ciphers for server-side SSL, and the servers might be configured to accept only RC4 ciphers. In this case, the SSL handshake between the SSL profile and the server fails because there are no common ciphers enabled. This results in the client connection being closed. If the client is using a browser, the user is likely to receive an error message indicating that the web page failed to load.

You can specify a string to indicate the available list of SSL ciphers, or you can use the default cipher string, DEFAULT. For Client SSL profiles, the definition of the DEFAULT cipher string is ALL:!SSLv2:@SPEED. For Server SSL profiles, the definition of the DEFAULT cipher string is COMPAT+HW:@SPEED.

The ciphers that the BIG-IP system supports are:

- SSLv2
  Note: We recommend that you refrain from using the SSLv2 cipher unless absolutely necessary.
- SSLv3
- TLSv1
- SGC/Set-up
• All standard protocol extensions and ciphers described in RFC 2246
• AES ciphers (described in RFC 3268)

◆ Note
The BIG-IP system supports the cipherlist format of OpenSSL version 0.9.7.

◆ Tip
In addition to specifying ciphers in an SSL profile, you can insert cipher specifications into the header of an HTTP request and then direct traffic based on those ciphers. For more information, see the web site http://devcentral.f5.com.

Configuring workarounds

OpenSSL supports a set of defect workarounds and SSL options. You can enable these workarounds and options as settings of an individual client-side or server-side SSL profile. The default value for the Options setting is Options List. Retaining the default value enables one option, which is Don’t insert empty fragments.

Table 9.3 lists and describes the possible workarounds and options that you can configure for an SSL profile.

<table>
<thead>
<tr>
<th>SSL Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Bugfixes Enabled</td>
<td>This option enables all of the defect workarounds that this table describes. It is usually safe to use the All Bugfixes Enabled option to enable the defect workaround options when you want compatibility with broken implementations.</td>
</tr>
<tr>
<td>Cipher server preference</td>
<td>When the BIG-IP system chooses a cipher, this option uses the server’s preferences instead of the client preferences. When this option is not set, the SSL server always follows the client’s preferences. When this option is set, the SSLv3/TLSv1 server chooses by using its own preferences. Due to the different protocol, for SSLv2 the server sends its list of preferences to the client, and the client always chooses the cipher.</td>
</tr>
<tr>
<td>Don’t insert empty fragments</td>
<td>This option disables a countermeasure against a SSL 3.0/TLS 1.0 protocol vulnerability affecting CBC ciphers. These ciphers cannot be handled by certain broken SSL implementations. This option has no effect for connections using other ciphers. This is the default value for the Enabled Options list.</td>
</tr>
</tbody>
</table>

Table 9.3 Workarounds and other SSL options
<table>
<thead>
<tr>
<th>SSL Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ephemeral RSA</td>
<td>This option uses ephemeral (temporary) RSA keys when doing RSA operations. According to the specifications, this is only done when an RSA key can only be used for signature operations (namely under export ciphers with restricted RSA key length). By setting this option, the BIG-IP system always uses ephemeral RSA keys. This option breaks compatibility with the SSL/TLS specifications and can lead to interoperability problems with clients, and we therefore do not recommend it. You should use ciphers with EDH (ephemeral Diffie-Hellman) key exchange instead. This option is ignored for server-side SSL.</td>
</tr>
<tr>
<td>Microsoft session ID bug</td>
<td>This option handles a Microsoft® session ID problem.</td>
</tr>
<tr>
<td>Netscape CA DN bug workaround</td>
<td>This option handles a defect regarding system instability. If the system accepts a Netscape® browser connection, demands a client cert, has a non-self-signed CA that does not have its CA in Netscape, and the browser has a certificate, then the system crashes or hangs.</td>
</tr>
<tr>
<td>Netscape challenge bug</td>
<td>This option handles the Netscape challenge problem.</td>
</tr>
<tr>
<td>Netscape demo cipher change bug workaround</td>
<td>This option deliberately manipulates the SSL server session resumption behavior to mimic that of certain Netscape servers (see the <a href="#">Netscape reuse cipher change bug workaround</a> description). We do not recommend this option for normal use and it is ignored for server-side SSL processing.</td>
</tr>
<tr>
<td>Netscape reuse cipher change bug workaround</td>
<td>This option handles a defect within Netscape-Enterprise/2.01, only appearing when connecting through SSLv2/v3 then reconnecting through SSLv3. In this case, the cipher list changes. First, a connection is established with the RC4-MD5 cipher list. If it is then resumed, the connection switches to using the DES-CBC3-SHA cipher list. However, according to RFC 2246, (section 7.4.1.3, cipher_suite) the cipher list should remain RC4-MD5. As a workaround, you can attempt to connect with a cipher list of DES-CBC-SHA:RC4-MD5 and so on. For some reason, each new connection uses the RC4-MD5 cipher list, but any re-connection attempts to use the DES-CBC-SHA cipher list. Thus Netscape, when reconnecting, always uses the first cipher in the cipher list.</td>
</tr>
<tr>
<td>No SSLv2</td>
<td>Do not use the SSLv2 protocol.</td>
</tr>
<tr>
<td>No SSLv3</td>
<td>Do not use the SSLv3 protocol.</td>
</tr>
<tr>
<td>No session resumption on renegotiation</td>
<td>When the BIG-IP system performs renegotiation as an SSL server, this option always starts a new session (that is, session resumption requests are only accepted in the initial handshake). The system ignores this option for server-side SSL processing.</td>
</tr>
<tr>
<td>No TLSv1</td>
<td>Do not use the TLSv1 protocol.</td>
</tr>
<tr>
<td>Microsoft big SSLv3 buffer</td>
<td>This option enables a workaround for communicating with older Microsoft® applications that use non-standard SSL record sizes.</td>
</tr>
</tbody>
</table>

*Table 9.3 Workarounds and other SSL options*
### Table 9.3 Workarounds and other SSL options

<table>
<thead>
<tr>
<th>SSL Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft IE SSLv2 RSA padding</td>
<td>This option enables a workaround for communicating with older Microsoft® applications that use non-standard RSA key padding. This option is ignored for server-side SSL.</td>
</tr>
<tr>
<td>Passive close</td>
<td>When set to Default, this option enables the All Bugfixes Enabled option. When set to None, this option disables all workarounds. We do not recommend setting this option to None.</td>
</tr>
<tr>
<td>PKCS1 check 1</td>
<td>This debugging option deliberately manipulates the PKCS1 padding used by SSL clients in an attempt to detect vulnerability to particular SSL server vulnerabilities. We do not recommend this option for normal use. The system ignores this option for client-side SSL processing.</td>
</tr>
<tr>
<td>PKCS1 check 2</td>
<td>This debugging option deliberately manipulates the PKCS1 padding used by SSL clients in an attempt to detect vulnerability to particular SSL server vulnerabilities. We do not recommend this option for normal use. The system ignores this option for client-side SSL processing.</td>
</tr>
<tr>
<td>Single DH use</td>
<td>This option creates a new key when using temporary/ephemeral DH parameters. You must use this option if you want to prevent small subgroup attacks, when the DH parameters were not generated using strong primes (for example, when using DSA-parameters). If strong primes were used, it is not strictly necessary to generate a new DH key during each handshake, but we do recommend this. You should enable the Single DH use option whenever temporary/ephemeral DH parameters are used.</td>
</tr>
<tr>
<td>SSLEAY 080 client DH bug workaround</td>
<td>This option enables a workaround for communicating with older SSLeay-based applications that specify an incorrect Diffie-Hellman public value length. This option is ignored for server-side SSL.</td>
</tr>
<tr>
<td>SSL Ref2 reuse cert type bug</td>
<td>This option handles the SSL re-use certificate type problem.</td>
</tr>
<tr>
<td>TLS D5 bug workaround</td>
<td>This option is a workaround for communicating with older TLSv1-enabled applications that specify an incorrect encrypted RSA key length. This option is ignored for server-side SSL.</td>
</tr>
<tr>
<td>TLS block padding bug workaround</td>
<td>This option enables a workaround for communicating with older TLSv1-enabled applications that use incorrect block padding.</td>
</tr>
<tr>
<td>TLS rollback bug workaround</td>
<td>This option disables version rollback attack detection. During the client key exchange, the client must send the same information about acceptable SSL/TLS protocol levels as it sends during the first hello. Some clients violate this rule by adapting to the server’s answer. For example, the client sends an SSLv2 hello and accepts up to SSLv3.1 (TLSv1), but the server only understands up to SSLv3. In this case, the client must still use the same SSLv3.1 (TLSv1) announcement. Some clients step down to SSLv3 with respect to the server’s answer and violate the version rollback protection. This option is ignored for server-side SSL.</td>
</tr>
</tbody>
</table>
Note that when configuring protocol versions, you must ensure that the protocol versions configured for the BIG-IP system match those of the system’s peer. That is, protocol versions specified in the client-side SSL profile must match those of the client, and protocol versions specified in the server-side SSL profile must match those of the server. Thus, for both client-side and server-side SSL connections, you can specify the protocol versions that you do not want the BIG-IP system to allow.

You can declare up to two of the three protocol versions to be invalid: SSLv2, SSLv3, and TLSv1. If no protocol versions are specified, the BIG-IP system allows all SSL protocol versions.

◆ Note

We recommend that at a minimum you specify protocol version SSLv2 as invalid.

To specify workaround options, locate the **Options** setting and verify that it is set to **Options List** (the default value). From the **Options List** setting, select any of the options you wish to configure, and click the **Enable** button. When you are finished configuring the SSL profile, click the **Finished** button.

### Enabling ModSSL method emulation

This setting enables or disables ModSSL method emulation. You enable this setting when the OpenSSL methods are inadequate.

When you enable the **ModSSL Methods** setting, you can then write an iRule, using the `HTTP::header insert_modssl_fields` command, which inserts some of the ModSSL options as headers into HTTP requests. The options that you can insert into an HTTP request are listed in Table 9.4.

<table>
<thead>
<tr>
<th>Header Type</th>
<th>Header Name and Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate status</td>
<td>SSLClientCertStatus: [status]</td>
<td>The status of the client certificate. The value of [status] can be NoClientCert, OK, or Error. If status is NoClientCert, only this header is inserted into the request. If status is Error, the error is followed by a numeric error code.</td>
</tr>
<tr>
<td>Certificate serial number</td>
<td>SSLClientCertSerialNumber: [serial]</td>
<td>The serial number of the certificate.</td>
</tr>
<tr>
<td>Signature algorithm of the certificate</td>
<td>SSLClientCertSignatureAlgorithm: [alg]</td>
<td>The signature algorithm of the certificate.</td>
</tr>
<tr>
<td>Issuer of the certificate</td>
<td>SSLClientCertIssuer: [issuer]</td>
<td>The issuer of the certificate.</td>
</tr>
</tbody>
</table>

*Table 9.4* ModSSL options for use with iRules
Configuring the SSL session cache

You can configure timeout and size values for the SSL session cache. Because each profile maintains a separate SSL session cache, you can configure the values on a per-profile basis.

Setting SSL session cache size

Using the Configuration utility, you can specify the maximum size of the SSL session cache. The default value for the size of the SSL session cache is 20,000 entries. A value of 0 disallows session caching.

You configure the values for the maximum size of the session cache on a per-profile basis. To specify an SSL session cache size, locate the Cache Size setting and retain the default cache-size value or type a new value.

Setting SSL session cache timeout

Using the Configuration utility, you can specify the number of usable lifetime seconds of negotiated SSL session IDs. The default timeout value for the SSL session cache is 300 seconds. Acceptable values are integers greater than or equal to 5.

Clients attempting to resume an SSL session with an expired session ID are forced to negotiate a new session.

◆ WARNING

If the timeout value for the client-side SSL session cache is set to zero, the SSL session IDs negotiated with that profile’s clients remain in the session cache until the cache is filled and the purging of entries begins. Setting a value of zero can introduce a significant security risk if valuable resources...
are available to a client that is reusing those session IDs. It is therefore common practice to set the SSL session cache timeout to a length of time no greater than 24 hours, and for significantly shorter periods.

To specify an SSL session cache timeout, locate the Cache Timeout setting, and retain the default value or, from the list, select Specify, Immediate, or Indefinite. If you select Specify, type a value.

Specifying an alert timeout value

The Alert Timeout setting specifies the duration in seconds that the BIG-IP system waits while trying to close an SSL connection, before the connection is reset. The default timeout value for this setting is 60 seconds. To specify an alert timeout, locate the Alert Timeout setting, and retain the default value or, from the list, select Specify or Indefinite. If you select Specify, type a value.

Specifying a handshake timeout value

The Handshake Timeout setting specifies the amount of time in seconds that the BIG-IP system spends attempting to perform an SSL handshake. The default timeout value for this setting is 60 seconds. To specify an alert timeout, locate the Handshake Timeout setting, and retain the default value or, from the list, select Specify or Indefinite. If you select Specify, type a value.

Forcing renegotiation of SSL sessions

Long-lived connections are susceptible to man-in-the-middle attacks. To prevent such attacks, you can force the BIG-IP system to renegotiate SSL sessions, based on either time period or application size. You can also force the BIG-IP system to terminate an SSL session after receiving a specified number of records.

Renegotiating sessions based on a time period

The Renegotiate Period setting specifies the number of seconds from the initial connect time that the system renegotiates an SSL session. The options are a number you specify, indefinite, and default. The default is indefinite, meaning that you do not want the system to renegotiate SSL sessions. Each time the session renegotiation is successful, essentially a new connection is started. Therefore, the system attempts to renegotiate the session again in the specified amount of time following the successful session renegotiation. For example, setting the renegotiate period to 3600 seconds triggers session renegotiation at least once an hour.

To specify a renegotiation period, locate the Renegotiate Period setting and retain the default value, or select Specify. If selecting Specify, type a value.
Renegotiating sessions based on application data size

The **Renegotiate Size** setting forces the BIG-IP system to renegotiate an SSL session after the specified number of megabytes of application data have been transmitted over the secure channel. The default value for this setting is **Indefinite**.

To specify a renegotiation size, locate the **Renegotiate Size** setting and retain the default value, or type a new value.

Specifying the maximum record delay

While the BIG-IP system waits for the client to initiate a renegotiation, the **Renegotiate Max Record Delay** setting forces the BIG-IP system to terminate an SSL session after receiving the specified maximum number of SSL records. If the BIG-IP system receives more than the maximum number of SSL records, it closes the connection. The default value for this setting, in seconds, is **10**.

To specify a maximum record delay, locate the **Renegotiate Max Record Delay** setting and retain the default value or type a new value.

Configuring SSL shutdowns

With respect to the shutdown of SSL connections, you can configure two settings on the BIG-IP system: **Unclean Shutdown** and **Strict Resume**.

Disabling unclean SSL shutdowns

In an **unclean shutdown**, underlying TCP connections are closed without exchanging the required SSL shutdown alerts. However, you use the **Unclean Shutdown** setting to disable unclean shutdowns and thus force the SSL profile to perform a clean shutdown of all SSL connections by configuring this setting.

This feature is especially useful with respect to the Internet Explorer browser. Different versions of the browser, and even different builds within the same version of the browser, handle shutdown alerts differently. Some versions or builds require shutdown alerts from the server, while others do not, and the SSL profile cannot always detect this requirement or lack of it. In the case where the browser expects a shutdown alert but the SSL profile has not exchanged one (the default setting), the browser displays an error message.

By default, this setting is enabled, which means that the BIG-IP system performs unclean shutdowns of all SSL connections. To disable unclean shutdowns, locate the **Unclean Shutdown** setting and clear the check box.
Discontinuing SSL sessions

Using the Strict Resume setting, you can configure the BIG-IP system to discontinue an SSL session after an unclean shutdown. By default, this setting is disabled, which causes the BIG-IP system to resume SSL sessions after an unclean shutdown. If you enable this setting, the BIG-IP system does not resume SSL sessions after an unclean shutdown.

To discontinue SSL sessions after an unclean shutdown, locate the Strict Resume setting and check the box.

Accepting non-SSL connections

Using the Non-SSL Connections setting, you can configure the BIG-IP system to accept connections that are not SSL connections. In this case, connections pass through the BIG-IP system in clear-text format. By default, this setting is disabled.
Configuring client and server authentication settings

This section describes the settings that appear in the Client Authentication section of a Client SSL Profile screen, or the Server Authentication section of a Server SSL Profile screen. For information on configuring the other SSL profile settings, see Configuring SSL profile settings, on page 9-13. For the procedure to create or modify an SSL profile, see Chapter 5, Understanding Profiles.

Table 9.5 lists and describes the authentication settings of a Client or Server SSL 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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client or Server Certificate</td>
<td>Configures the SSL profile to either request, require, or ignore certificates presented by a client or a server.</td>
<td>Ignore</td>
</tr>
<tr>
<td>Frequency</td>
<td>Specifies whether the profile should authenticate a client once per session, or once per session and upon each subsequent re-use of an SSL session. For more information, see Configuring per-session authentication, on page 9-27.</td>
<td>Once</td>
</tr>
<tr>
<td>Certificate Chain Traversal</td>
<td>Specifies the maximum number of certificates that can be traversed in a client certificate chain. For more information, see Configuring authentication depth, on page 9-28.</td>
<td>9</td>
</tr>
<tr>
<td>Advertised Certificate Authorities</td>
<td>Specifies the CAs that you would like to advertise to clients as being trusted by the profile. For more information, see Advertising a list of trusted client CAs, on page 9-28. This attribute applies to client-side profiles only.</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: This setting only appears when you set the Client Certificate setting to Request, Require, or Auto, or when you set the Server Certificate setting to Require.

Table 9.5 Authentication settings of an SSL profile
Configuring certificate presentation

By configuring the Client Certificate or Server Certificate setting, you can cause the BIG-IP system to handle authentication of clients or servers in certain ways.

For client-side processing, the possible values of the Client Certificate setting are:

- **Request**
  Request and verify a client certificate. In this case, the SSL profile always grants access regardless of the status or absence of the certificate.

- **Require**
  Require a client to present a valid and trusted certificate before granting access.

- **Ignore**
  Ignore a certificate (or lack of one) and therefore never authenticate the client. The ignore setting is the default setting, and when used, causes any per-session authentication setting to be ignored. For information on configuring per-session authentication, see Configuring per-session authentication, on page 9-27.

- **Auto**
  Ignore a client certificate until an authentication module requests one. In this case, the BIG-IP system initiates a mid-session SSL handshake, as though the option were set to Request. We recommend this setting only for those connections for which the presentation of client certificate is not required.

---

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate Name</td>
<td>Authenticates a target server based on the Common Name (CN) embedded in a server certificate. For more information, see Configuring name-based authentication, on page 9-29. This attribute applies to server-side profiles only. <strong>Note:</strong> This setting only appears when you set the Server Certificate setting to Require.</td>
<td>No default value</td>
</tr>
<tr>
<td>Certificate Revocation List (CRL)</td>
<td>Configures certificate revocation by maintaining a list of revoked client certificates. For more information, see Certificate revocation, on page 9-29. <strong>Note:</strong> This setting only appears when you set the Client Certificate setting to Request, Require, or Auto, or when you set the Server Certificate setting to Require.</td>
<td>No default values</td>
</tr>
</tbody>
</table>

*Table 9.5 Authentication settings of an SSL profile*
**WARNING**

If you are using the LDAP-based client authorization feature, use of the **Request** or **Ignore** options can sometimes cause a connection to terminate. For more information on LDAP-based client authorization, see Chapter 10, **Authenticating Application Traffic**.

**Tip**

The **Request** value works well with the header insertion feature. Configuring the SSL profile to insert client certificate information into an HTTP client request, and to authenticate clients based on the **Request** option, enables the BIG-IP system or a server to then perform actions such as redirecting the request to another server, sending different content back to the client, or performing client certificate or session ID persistence.

For server-side processing, the possible values of the **Server Certificate** setting are:

- **Require**
  Require a server to present a valid and trusted certificate before granting access.

- **Ignore**
  Ignore a certificate (or lack of one) and therefore never authenticate the server. The **Ignore** value is the default setting, and when used, causes any per-session authentication setting to be ignored. For information on configuring per-session authentication, see **Configuring per-session authentication**, following.

### Configuring per-session authentication

With the **Frequency** setting, you can configure an SSL profile to require authentication either once per SSL session (**once**), or once upon each subsequent re-use of an SSL session (**always**). The default setting for this option is **once**.

Whether you set this value to **once** or **always** depends on your application. A well-designed web application should only need to verify a certificate once per session. We recommend for performance reasons that you use the default setting (**once**) whenever possible.

You can modify the SSL profile to require authentication not only once per session, but also upon each subsequent re-use of an SSL session.
Advertising a list of trusted client CAs

For client-side profiles only, if you intend to configure the SSL profile to require or request client certificates for authentication, you will want the profile to send to clients a list of CAs that the server is likely to trust. You can do this by configuring the **Advertised Certificate Authorities** setting. This list, known as the Client Certificate CA file, is different from the client Trusted CAs file. This is because, in some cases, you might have a client that does not possess a valid client certificate, in which case you might not want to reveal the actual list of CAs that the profile trusts. The client certificate CA file solves this problem by allowing the profile to advertise a list of CAs that is different from the actual client trusted CAs file configured as part of certificate verification.

**Tip**

Although the contents of the Client Certificate CA file can differ from that of the Client Trusted CAs file, it is best, for compatibility reasons, to set the Client Certificate CA option to match the actual Client Trusted CAs file. This is because modern browsers might not permit SSL session negotiation to proceed if the peer that requests the client certificate does not provide a list of trusted CAs.

To configure the profile to send this list, you can specify a PEM-formatted certificate file that contains one or more CAs that a server trusts for client authentication. If no Client Certificate CA file is specified, no list of trusted CAs is sent to a client.

**Note**

The maximum size of native SSL handshake messages that the BIG-IP system allows is 14304 bytes. Consequently, if the SSL handshake is negotiating a native cipher and the total length of all messages in the handshake exceeds this byte threshold, the handshake can fail. Although typical use does not cause message length to exceed this threshold, we recommend that when configuring a Client SSL profile to request or require client certificates, you avoid specifying large numbers of certificates with the **Advertised Certificate Authorities** setting. This minimizes the number of certificates that must be exchanged during a Client SSL handshake.

Configuring authentication depth

Using the **Certificate Chain Traversal Depth** setting, you can configure the maximum number of certificates that can be traversed in the certificate chain. The default value is 9. If a longer chain is provided, and the client has not been authenticated within this number of traversals, client or server certificate verification fails. If the authentication depth value is set to zero, then only the client or server certificate, and one of the chain files, are examined.
Configuring name-based authentication

For server-side profiles only, the BIG-IP system supports name-based authentication, which guards against man-in-the-middle attacks. When you configure the Authenticate Name setting for a server-side profile, the BIG-IP system checks the name against the Common Name (CN) listed in the certificate that the target server presents to the BIG-IP system. If the name attribute that you specify does not match the CN in the server certificate, the BIG-IP system closes the connection. An example of a CN is www.f5.com.

Certificate revocation

The Certificate Revocation List (CRL) setting allows the BIG-IP system to use CRLs to check revocation status of a certificate prior to authenticating a client or server.

To configure CRLs for an SSL profile, you must configure a CRL file, which contains a list of revoked client or server certificates. When specifying a list of revoked certificates, the file that you specify must be a PEM-formatted file.

◆ Important

CRL files can become outdated, and might need to be updated as often as every day, or as seldom as every 30 days. If your CRL file is out-of-date, the BIG-IP system rejects all certificates, both valid and invalid. For this reason, it is important to keep your CRL files up-to-date at all times. You can do this by accessing the CRL in the /config/ssl/ssl.crl directory and then using the openssl crl command. For more information, see http://www.openssl.org/docs/.

As an alternative to using CRLs, you can use the Online Certificate Status Protocol (OCSP) feature, which ensures up-to-date information on certificate revocation status. For more information, see Chapter 10, Authenticating Application Traffic.
Authenticating Application Traffic

- Introducing remote authentication
- Implementing an LDAP authentication module
- Implementing a RADIUS authentication module
- Implementing a TACACS+ authentication module
- Implementing an SSL client certificate LDAP authentication module
- Implementing an SSL OCSP authentication module
- Implementing a CRLDP authentication module
- Implementing a Kerberos Delegation authentication module
Introducing remote authentication

A significant feature of the BIG-IP® local traffic management system is its ability to support Pluggable Authentication Module (PAM) technology. PAM technology allows you to choose from a number of different authentication and authorization schemes to use to authenticate or authorize network traffic.

The goal of PAM technology is to separate an application, such as the BIG-IP system, from its underlying authentication technology. This means that you can dictate the particular authentication/authorization technology that you want the BIG-IP system to use to authenticate application traffic coming into the BIG-IP system.

To this end, the BIG-IP system offers several authentication schemes, known as authentication modules. These authentication modules allow you to use a remote system to authenticate or authorize application requests that pass through the BIG-IP system.

To implement an authentication module, you use the Configuration utility to access the profile screens within the Local Traffic section of the utility. Using these screens, you can configure settings for the type of authentication module you want to implement.

◆ Note

*The BIG-IP system normally routes remote authentication traffic through a Traffic Management Microkernel (TMM) switch interface (that is, an interface associated with a VLAN and a self IP address), rather than through the management interface. Therefore, if the TMM service has been stopped for any reason, remote authentication is not available until the service is running again. For more information on routing traffic through BIG-IP interfaces, see the TMOSTM Management Guide for BIG-IP® Systems.*

BIG-IP system authentication modules

The BIG-IP system modules that you can implement for remote authentication are:

◆ Lightweight Directory Access Protocol (LDAP)
  The BIG-IP system can authenticate or authorize network traffic using data stored on a remote LDAP server or a Microsoft® Windows® Active Directory® server. Client credentials are based on basic HTTP authentication (user name and password).

◆ Remote Authentication Dial-In User Service (RADIUS)
  The BIG-IP system can authenticate network traffic using data stored on a remote RADIUS server. Client credentials are based on basic HTTP authentication (user name and password).
◆ **TACACS+**
The BIG-IP system can authenticate network traffic using data stored on a remote TACACS+ server. Client credentials are based on basic HTTP authentication (user name and password).

◆ **SSL client certificate LDAP**
The BIG-IP system can authorize network traffic using data stored on a remote LDAP server. Client credentials are based on SSL certificates, as well as defined user groups and roles.

◆ **Online Certificate Status Protocol (OCSP)**
The BIG-IP system can check on the revocation status of a client certificate using data stored on a remote OCSP server. Client credentials are based on SSL certificates.

### Implementing authentication modules

Using the BIG-IP system, you can implement any of the available authentication modules listed in the previous section. Implementing an authentication module requires you to create or configure the following objects:

◆ **A RADIUS server or SSL OCSP responder object**
This is required for RADIUS and SSL OCSP authentication modules only. Server objects and responder objects consist of settings pertaining to remote RADIUS servers or OCSP responders.

◆ **A configuration object**
It is the **configuration object** that controls the authentication module, through a group of configurable settings. Examples of configuration object settings are **Hosts**, **Service Port**, and **Search Time Limit**.

◆ **An authentication profile**
An **authentication profile** is an object that specifies the type of authentication module you want to implement, a parent profile, and the configuration object. You can either use the default profile that the BIG-IP system provides for each type of authentication module, or create a custom profile. For background information on profiles, see Chapter 5, **Understanding Profiles**.

◆ **An authentication iRule**
For any type of PAM module that you want to implement, the BIG-IP system provides a corresponding default authentication iRule that you must associate with your profile and your virtual server. An **authentication iRule** is a TCL-based script that performs the authentication/authorization action, according to the settings of the corresponding profile and configuration objects.

In most cases, you can use the default iRule, rather than creating a custom one. Creating custom iRules™ is necessary when you want to implement multiple authentication modules of the same type, such as multiple LDAP authentication modules, each with different configuration object and profile settings. For information on creating custom iRules, see Chapter 17, **Writing iRules**.
The process you use to implement an authentication module is a simple one. For example, to implement an LDAP authentication module, which uses a remote LDAP server to authenticate client traffic using user name and password credentials, you perform the following tasks. Note that this example creates a custom profile and uses the default iRule that the BIG-IP system provides.

**To implement an authentication module**

1. Create an authentication configuration object named `my_ldap_config`. For information on creating an LDAP authentication configuration object, see *Creating an LDAP configuration object*, on page 10-5.

2. Create a custom authentication profile named `my_ldap_profile`, in which you specify the authentication module type as LDAP, specify a parent profile (either the default `ldap` profile or another custom profile that you created), and reference the configuration object `my_ldap_config`. For information on creating an LDAP profile, see *Creating an LDAP profile*, on page 10-7.

3. Configure the virtual server to reference both the custom profile `my_ldap_profile` and the default authentication iRule `auth_ldap`. For information on how to configure virtual server settings, see Chapter 2, *Configuring Virtual Servers*. 
Implementing an LDAP authentication module

An LDAP authentication module is a mechanism for authenticating or authorizing client connections passing through a BIG-IP system. This module is useful when your authentication or authorization data is stored on a remote LDAP server or a Microsoft® Windows Active Directory server, and you want the client credentials to be based on basic HTTP authentication (that is, user name and password).

With the LDAP authentication module, the BIG-IP system can indicate that the authentication was a success or failure, or that the LDAP server needs a credential of some sort.

Additionally, the system can take some action based on certain information that the server returns in the LDAP query response. For example, LDAP response information can indicate the user’s group membership, or it can indicate that the user’s password has expired. To configure the BIG-IP system to return specific data in an LDAP response, you can write an iRule, using the commands AUTH::subscribe, AUTH::unsubscribe, and AUTH::result-data. For more information, see Chapter 17, Writing iRules, and the F5 Networks DevCentral web site, http://devcentral.f5.com.

To implement an LDAP authentication module, you must configure the BIG-IP system to access data on a remote LDAP server. To do this, you must create:

- An LDAP configuration object
- An LDAP profile

◆ Important

When you create LDAP objects and profiles, the BIG-IP system places them into your current administrative partition. The default profile always resides in partition Common. For information on partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

After you create these objects, you must then assign the LDAP profile to a virtual server.

◆ Note

If the Create button is unavailable in the Configuration utility when you attempt to create LDAP authentication objects, this indicates that your user role does not grant you permission to create the object.
Creating an LDAP configuration object

When you create an LDAP configuration object, you configure a variety of settings. Table 10.1 shows the settings and values that you configure for an LDAP configuration object. Note that this table groups the settings into the same categories that you see on the New Authentication Configuration screen. For the detailed procedure on how to create this object, see To create an LDAP configuration object, on page 10-6.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Properties</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to LDAP.</td>
<td>No default value</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote LDAP Tree</td>
<td>Specifies the search base distinguished name.</td>
<td>No default value</td>
</tr>
<tr>
<td>Hosts</td>
<td>Lists the addresses of the LDAP servers that the BIG-IP system uses to obtain authentication data.</td>
<td>No default value</td>
</tr>
<tr>
<td>Service Port</td>
<td>Specifies the port number for the LDAP service.</td>
<td>389 (non-SSL) / 636 (SSL-enabled)</td>
</tr>
<tr>
<td>LDAP Version</td>
<td>Specifies the version number of the LDAP application.</td>
<td>3</td>
</tr>
<tr>
<td>Bind DN</td>
<td>Specifies the distinguished name of an account to which to bind, in order to perform searches. This search account is a read-only account used to do searches. The admin account can be used as the search account. If no admin DN is specified, then no bind is attempted. This setting is only required when a site does not allow anonymous searches. If the remote server is a Microsoft® Windows Active Directory server, the distinguished name must be in the form of an email address.</td>
<td>No default value</td>
</tr>
<tr>
<td>Bind Password</td>
<td>Specifies the password for the search account created on the LDAP server. This setting is required if you use a bind DN.</td>
<td>No default value</td>
</tr>
<tr>
<td>Confirm Bind Password</td>
<td>Confirms the password for the bind distinguished name. This setting is optional.</td>
<td>No default value</td>
</tr>
<tr>
<td>Search Time Limit</td>
<td>Specifies a time limit for a search.</td>
<td>30</td>
</tr>
<tr>
<td>Bind Time Limit</td>
<td>Specifies a Bind time limit.</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 10.1 Settings of an LDAP configuration object
To create an LDAP configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.

2. From the Authentication menu, choose Configurations.

3. In the upper right corner of the screen, click Create.
   This displays the New Configuration screen.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Specifies a filter. This setting is used for authorizing client traffic.</td>
<td>No default value</td>
</tr>
<tr>
<td>Login Attribute</td>
<td>Specifies a login attribute. Normally, the value for this setting is uid; however, if the server is a Microsoft Windows Active Directory server, the value must be the account name sAMAccountName (case-insensitive).</td>
<td>No default value</td>
</tr>
<tr>
<td>Group DN</td>
<td>Specifies the group distinguished name. This setting is used for authorizing client traffic.</td>
<td>No default value</td>
</tr>
<tr>
<td>Group Member Attribute</td>
<td>Specifies an group member attribute. This setting is used for authorizing client traffic.</td>
<td>No default value</td>
</tr>
<tr>
<td>SSL</td>
<td>Allowed values are: Enabled, Disabled, and Start TLS. Note that when enabled, the BIG-IP system changes the service port number from 389 to 636.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Check SSL Peer</td>
<td>When enabled, checks an SSL peer.</td>
<td>Disabled</td>
</tr>
<tr>
<td>SSL CA Certificate</td>
<td>Specifies the name of an SSL CA certificate. Allowed values are: None, Default, and Specify Full Path.</td>
<td>None</td>
</tr>
<tr>
<td>SSL Client Key</td>
<td>Specifies the name of an SSL client key. Allowed values are: None, Default, and Specify Full Path.</td>
<td>None</td>
</tr>
<tr>
<td>SSL Client Certificate</td>
<td>Specifies the name of an SSL client certificate. Allowed values are: None, Default, and Specify Full Path.</td>
<td>None</td>
</tr>
<tr>
<td>SSL Ciphers</td>
<td>Specifies SSL ciphers.</td>
<td>No default value</td>
</tr>
<tr>
<td>Ignore Unknown User</td>
<td>When enabled, causes the BIG-IP system to ignore unknown users.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Warning Logging</td>
<td>Enables or disables warning messages.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Debug Logging</td>
<td>Enables or disables SYSLOG debugging information at LOG_DEBUG level. Not recommended for normal use.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Table 10.1 Settings of an LDAP configuration object
4. For the **Name** setting, type a unique name for the configuration object, such as `my_ldap_config`.
   
   **Note**: The name must consist of lowercase characters only.

5. For the **Type** setting, select **LDAP**.
   
   The screen expands to show several settings.

6. Modify or retain values for all settings shown.
   
   (To configure advanced settings, locate the **Configuration** heading and select **Advanced**.)

7. Click **Finished**.

### Creating an LDAP profile

Once you have created an LDAP configuration object, you must create or configure an LDAP profile. You do this by modifying the default `ldap` profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object that you have created.

In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create an LDAP profile, you configure some settings. Table 10.2 shows the settings and values that make up an LDAP profile. For the detailed procedure on creating an LDAP profile, see *To modify the default LDAP profile*, following, or *To create a custom LDAP profile*, on page 10-8.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to <strong>LDAP</strong>.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which you want to inherit values.</td>
<td><code>ldap</code></td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies whether the profile is enabled or disabled. Possible settings are <strong>Auto</strong>, <strong>Enabled</strong>, and <strong>Disabled</strong>.</td>
<td><strong>Enabled</strong></td>
</tr>
<tr>
<td>Configuration</td>
<td>Specifies an existing LDAP configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

*Table 10.2 Settings of an LDAP profile*
To modify the default LDAP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Profiles. This displays the list of default authentication profiles.
3. In the Name column, click ldap.
4. For the Mode setting, select Enabled or Auto.
5. For the Configuration setting, select a configuration object from the list. Note that None is not an allowed value.
6. For the Rule setting, specify an authentication iRule:
   - If you want to use the default iRule auth_ldap, leave the setting as is.
   - If you do not want to use the default iRule auth_ldap, select the name of an existing iRule that you have created.
7. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.
8. Click Finished.

To create a custom LDAP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Profiles.
3. In the upper right corner of the screen, click Create. This displays the New Profile screen.
4. For the Name setting, type a unique name for the profile, such as my_ldap_profile.
   Note: The name must consist of lowercase characters only.
5. For the **Type** setting, select **LDAP**.
   The screen expands to show additional settings.

6. For the **Parent Profile** setting, specify a type of profile:
   - If you want to use the default profile **ldap** for the parent profile, leave the setting as is.
   - If you want to use a custom profile for the parent profile, select a custom profile name from the list.

7. For the **Mode** setting, select **Enabled** or **Auto**.

8. For the **Configuration** setting, select a configuration object from the list.

9. For the **Rule** setting, specify an authentication iRule:
   - If you want to use the default iRule **auth_ldap**, leave the setting as is.
   - If you do not want to use the default iRule **auth_ldap**, select the name of an existing iRule that you have created.

10. For the **Idle Timeout** setting, retain the default value, specify a new value, or select **Indefinite**.

11. Click **Finished**.

After you have created an LDAP configuration object and an LDAP profile, you must assign the profile to the virtual server by configuring the virtual server’s **Authentication Profile** setting. For information on how to configure virtual server settings, see Chapter 2, *Configuring Virtual Servers*. 
Implementing a RADIUS authentication module

A RADIUS authentication module is a mechanism for authenticating client connections passing through a BIG-IP system. You use this module when your authentication data is stored on a remote RADIUS server. In this case, client credentials are based on basic HTTP authentication (that is, user name and password).

To implement a RADIUS authentication module, you must configure the BIG-IP system to access data on a remote RADIUS server. To do this, you must create:

- One or more high-level RADIUS server objects
- A RADIUS configuration object
- A RADIUS profile

Important

When you create RADIUS objects and profiles, the BIG-IP system places them into your current administrative partition. The default profile always resides in partition Common. For information on partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

After you create these objects, you must then assign the RADIUS profile to a virtual server.

Note

If the Create button is unavailable in the Configuration utility when you attempt to create RADIUS authentication objects, this indicates that your user role does not grant you permission to create the object.

Creating a RADIUS server object

When you create a RADIUS server object, you configure some settings. Table 10.3 shows the settings and values that make up a default RADIUS server object. For the detailed procedure on creating a server object, see To create a RADIUS server object, on page 10-11.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the RADIUS server object, such as my_radius_object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Host</td>
<td>Specifies a host name or IP address for the RADIUS server. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Service Port</td>
<td>Specifies the port for RADIUS authentication traffic.</td>
<td>1812</td>
</tr>
</tbody>
</table>

Table 10.3 Settings of a RADIUS sever definition
To create a RADIUS server object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.
2. From the Authentication menu, choose RADIUS Servers. This displays the RADIUS Server List screen.
3. In the upper right corner of the screen, click Create.
4. For the Name setting, type a unique name for the RADIUS server object, such as my_radius_server.
   Note: The name must consist of lowercase characters only.
5. For the Host setting, type a host name or IP address for the remote RADIUS server.
6. For the Secret and Confirm Secret settings, type the RADIUS secret.
7. Retain or modify all other settings.
8. Click Finished.
9. For redundant RADIUS servers, repeat these steps to create additional server objects.

Creating a RADIUS configuration object

When you create a RADIUS configuration object, you configure a variety of settings. Table 10.4, on page 10-12, shows the settings and values that make up a RADIUS configuration object. Note that this table groups the settings into the same categories that you see on the New Authentication Configuration screen. For the detailed procedure on how to create this configuration object, see To create a RADIUS configuration object, on page 10-12.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret</td>
<td>Sets the secret key used to encrypt and decrypt packets sent or received from the server. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Confirm Secret</td>
<td>Confirms the secret key supplied for the Secret setting. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Timeout</td>
<td>Specifies a timeout value.</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 10.3 Settings of a RADIUS sever definition
To create a RADIUS configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.
2. From the Authentication menu, choose Configurations.
3. In the upper right corner of the screen, click Create.
   This displays the New Configuration screen.
4. For the Name setting, specify a unique name for the configuration object, such as my_radius_config.
   Note: The name must consist of lowercase characters only.
5. For the Type setting, select RADIUS.
   The screen expands to show several settings.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to RADIUS.</td>
<td>No default value</td>
</tr>
<tr>
<td>RADIUS Servers</td>
<td>Lists the IP addresses of the RADIUS servers that the BIG-IP system will use to obtain authentication data. Note that for each server listed, you must create a corresponding RADIUS server definition. A RADIUS server definition specifies the server name, port number, RADIUS secret, and timeout value. For more information, see Table 10.3, on page 10-10.</td>
<td>No default value</td>
</tr>
<tr>
<td>Client ID</td>
<td>Sends a NAS-Identifier RADIUS attribute with string bar. If you do not specify a value for the Client ID setting, the PAM service type is used instead. This feature can be disabled by specifying a blank client ID.</td>
<td>No default value</td>
</tr>
<tr>
<td>Debug Logging</td>
<td>Enables SYSLOG debugging information at LOG_DEBUG level. We do not recommend this for normal use.</td>
<td>Disable</td>
</tr>
<tr>
<td>Accounting Bug</td>
<td>Disables validation of the accounting response vector. This option should only be necessary on older servers.</td>
<td>Disable</td>
</tr>
<tr>
<td>Retries</td>
<td>Specifies the number of authentication retries that the BIG-IP system allows before authentication fails.</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 10.4 Settings of a RADIUS configuration object
6. Modify or retain values for all settings shown.
   (To configure advanced settings, locate the Configuration heading and select Advanced.)

7. Click Finished.

Creating a RADIUS profile

Once you have created a RADIUS configuration object, you must create or configure a RADIUS profile. You do this by modifying the default radius profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object.

In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create a RADIUS profile, you configure a variety of settings. Table 10.5 shows the settings and values that make up a RADIUS profile. For the detailed procedure on creating a RADIUS profile, see To modify the default RADIUS profile, following, or To create a custom RADIUS profile, on page 10-14.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to RADIUS.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which you want to inherit values.</td>
<td>radius</td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies whether the profile is enabled or disabled. Possible settings are Auto, Enabled, and Disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Configuration</td>
<td>Specifies an existing RADIUS configuration object.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

Table 10.5 Settings of a RADIUS profile
To modify the default RADIUS profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Profiles. This displays the list of default authentication profiles.
3. In the Name column, click radius.
4. For the Mode setting, select Enabled or Auto.
5. For the Configuration setting, select a configuration object from the list. Note that None is not an allowed setting.
6. For the Rule setting, specify an authentication iRule:
   - If you want to use the default iRule auth_radius, leave the setting as is.
   - If you do not want to use the default iRule auth_radius, select the name of an existing iRule that you have created.
7. For theIdle Timeout setting, retain the default value, specify a new value, or select Indefinite.
8. Click Finished.

To create a custom RADIUS profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Profiles.
3. In the upper right corner of the screen, click Create. This displays the New Profile screen.
4. In the Name setting, type a unique name for the RADIUS profile, such as my_radius_profile.
   Note: The name must consist of lowercase characters only.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td>Specifies the name of an existing authentication rule. If you do not specify an iRule, the BIG-IP system uses the corresponding default iRule.</td>
<td>auth_radius</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the duration in seconds that an authentication or authorization request is idle before timing out. You can use the default value, specify a different value, or select Indefinite. For general information on timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 10.5 Settings of a RADIUS profile
5. In the **Type** setting, select **RADIUS**.  
The screen expands to show additional settings

6. For the **Parent Profile** setting, specify a type of profile:
   - If you want to use the default profile **radius** for the parent profile, leave the setting as is.
   - If you want to use a custom profile for the parent profile, select a custom profile name from the list.

7. In the **Mode** setting, select **Enabled** or **Auto**.

8. In the **Configuration** setting, select a configuration object from the list.

9. For the **Rule** setting, specify an authentication iRule:
   - If you want to use the default iRule **auth_radius**, leave the setting as is.
   - If you do not want to use the default iRule **auth_radius**, select the name of an existing iRule that you have created.

10. For the **Idle Timeout** setting, retain the default value, specify a new value, or select **Indefinite**.

11. Click **Finished**.

After you have created a RADIUS server object, a RADIUS configuration object, and a RADIUS profile, you must assign the profile to the virtual server by configuring the virtual server’s **Authentication Profile** setting.

For information on how to configure virtual server settings, see Chapter 2, *Configuring Virtual Servers*. 
Implementing a TACACS+ authentication module

A TACACS+ authentication module is a mechanism for authenticating client connections passing through a BIG-IP system. You use this module when your authentication data is stored on a remote TACACS+ server. In this case, client credentials are based on basic HTTP authentication (that is, user name and password).

To implement the TACACS+ authentication module, you must configure the BIG-IP system to access data on a remote TACACS+ server. To do this, you must create:

- A TACACS+ configuration object
- A TACACS+ profile

Important

When you create TACACS+ objects and profiles, the BIG-IP system places them into your current administrative partition. The default profile always resides in partition Common. For information on partitions, see the TMOS™ Management Guide for BIG-IP® Systems.

After you create these objects, you must then assign the TACACS+ profile to a virtual server.

Note

If the Create button is unavailable in the Configuration utility when you attempt to create TACACS+ authentication objects, this indicates that your user role does not grant you permission to create the object.

Creating a TACACS+ configuration object

When you create a TACACS+ configuration object, you configure a variety of settings. Table 10.6, on page 10-17, shows the settings and values that make up a TACACS+ configuration object. Note that this table groups the settings into the same categories that you see on the New Authentication Configuration screen. For the detailed procedure on how to create this object, see To create a TACACS+ configuration object, on page 10-18.
### Table 10.6 Settings of a TACACS+ configuration object

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to TACACS+.</td>
<td>No default value</td>
</tr>
<tr>
<td>Servers</td>
<td>Specifies a host name or IP address for the TACACS++ server. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Secret</td>
<td>Sets the secret key used to encrypt and decrypt packets sent or received from the server. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Confirm Secret</td>
<td>Confirms the secret key supplied for the Secret setting. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Encryption</td>
<td>Enables or disables encryption of TACACS+ packets. Recommended for normal use.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Service Name</td>
<td>Specifies the name of the service that the user is requesting to be authenticated to use. Identifying what the user is asking to be authenticated for enables the TACACS+ server to behave differently for different types of authentication requests. Specifying this setting is required. You can use following values: slip, ppp, arap, shell, tty-daemon, connection, system, and firewall.</td>
<td>No default value</td>
</tr>
<tr>
<td>Protocol Name</td>
<td>Specifies the protocol associated with the value specified in the Service Name setting, which is a subset of the associated service being used for client authorization or system accounting. You can use the following values: lcp, ip, ipx, atalk, vines, lat, xremote, tn3270, telnet, rlogin, pad, vpdn, ftp, http, deccp, osicp, and unknown.</td>
<td>No default value</td>
</tr>
<tr>
<td>Authentication</td>
<td>Specifies authentication options. Possible values are Authenticate to first server and Authenticate to each server until success.</td>
<td>Authenticate to first server</td>
</tr>
<tr>
<td>Accounting Information</td>
<td>If multiple TACACS+ servers are defined and PAM session accounting is enabled, sends accounting start and stop packets to the first available server or to all servers. Possible values are Send to first available server and Send to all servers.</td>
<td>Send to first available server</td>
</tr>
<tr>
<td>Debug Logging</td>
<td>Enables SYSLOG debugging information at LOG_DEBUG level. Not recommended for normal use.</td>
<td>Disable</td>
</tr>
</tbody>
</table>
To create a TACACS+ configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.
2. From the Authentication menu, choose Configurations.
3. In the upper right corner of the screen, click Create.
   This displays the New Configuration screen.
4. For the Name setting, specify a unique name for the configuration object, such as my_tacacs_config.
   Note: The name must consist of lowercase characters only.
5. For the Type setting, select TACACS+.
   The screen expands to show several settings.
6. Modify or retain values for all settings shown.
   (To configure advanced settings, locate the Configuration heading and select Advanced.)
7. Click Finished.

Creating a TACACS+ profile

Once you have created a TACACS+ configuration object, you must create or configure a TACACS+ profile. You do this by modifying the default tacacs+ profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object.

In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create a TACACS+ profile, you configure a variety of settings. Table 10.7, on page 10-19, shows the settings and values that make up a TACACS+ profile. For the detailed procedure on creating a TACACS+ profile, see To modify the default TACACS+ profile, following, or To create a custom TACACS+ profile, on page 10-20.
To modify the default TACACS+ profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.

2. From the Authentication menu, choose Profiles.
   This displays the list of default authentication profiles.

3. In the Name column, click tacacs+.

4. For the Mode setting, select Enabled or Auto.

5. For the Configuration setting, select a configuration object from the list. Note that None is not an allowed setting.

6. For the Rule setting, specify an authentication iRule:
   • If you want to use the default iRule auth_tacacs, leave the setting as is.
   • If you do not want to use the default iRule auth_tacacs, select the name of an existing iRule that you have created.

7. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.

8. Click Finished.
To create a custom TACACS+ profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.

2. From the Authentication menu, choose Profiles.

3. In the upper right corner of the screen, click Create.
   This displays the New Profile screen.

4. For the Name setting, specify a unique name for the profile.
   Note: The name must consist of lowercase characters only.

5. For the Type setting, select TACACS+.

6. For the Parent Profile setting, specify a type of profile:
   • If you want to use the default profile tacacs for the parent profile, leave the setting as is.
   • If you want to use a custom profile for the parent profile, select a custom profile name from the list.

7. For the Mode setting, select Enabled or Auto.

8. For the Configuration setting, select a configuration object from the list.

9. For the Rule setting, specify an authentication iRule:
   • If you want to use the default iRule auth_tacacs, leave the setting as is.
   • If you do not want to use the default iRule auth_tacacs, select the name of an existing iRule that you have created.

10. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.

11. Click Finished.

After you have created a TACACS+ configuration object and a TACACS+ profile, you must assign the profile to the virtual server by configuring the virtual server’s Authentication Profile setting.

For information on how to configure virtual server settings, see Chapter 2, Configuring Virtual Servers.
Implementing an SSL client certificate LDAP authentication module

An SSL client certificate LDAP authentication module is a mechanism for authorizing client connections passing through a BIG-IP system. With the SSL client certificate LDAP authentication module, you can use a remote LDAP server to impose access control on application traffic. The module bases this access control on SSL certificates, as well as user groups and roles that you specify.

With the SSL client certificate LDAP authentication module, the BIG-IP system can indicate that the authorization was a success or failure, or that the LDAP server needs a credential of some sort.

Additionally, the system can take some action based on certain information that the server returns in the LDAP query response. For example, LDAP response information can indicate the user's group membership, or it can indicate that the user's password has expired. To configure the BIG-IP system to return specific data in an LDAP response, you can write an iRule, using the commands AUTH::subscribe, AUTH::unsubscribe, and AUTH::result-data. For more information, see Chapter 17, Writing iRules, and the F5 Networks DevCentral web site, http://devcentral.f5.com.

◆ Important

Prior to implementing an SSL client certificate LDAP authentication module, you must configure and implement a Client SSL profile. When you configure a Client SSL profile, we recommend that you set the Client Certificate property to either Require or Auto. Setting this property to Ignore or Request could cause SSL connections to terminate. For a description of the Client Certificate setting, see Chapter 9, Managing SSL Traffic.

Understanding SSL client certificate authorization

With SSL client certificate LDAP authorization, the BIG-IP system can authorize clients based on signed client certificates issued by trusted CAs. Then, to further enhance the ability of the system to authorize client requests, you can also specify groups and roles. Basing authorization on certificates as well as groups and roles provides the flexibility you need to control client access to system resources.

Before you can implement an SSL client certificate LDAP module, you must understand the two different types of credentials that the BIG-IP system uses to authorize application traffic using data on a remote LDAP server. These two types of credentials are:

- SSL certificates
- Groups and roles
Using SSL certificates for LDAP authorization

During the process of authorizing a client, the BIG-IP system must search the LDAP database. When using certificate-based authorization, the system can search the LDAP database in three ways:

- **User**
  If certificates are not stored in the LDAP database, you can configure the system to extract a user name from the certificate presented as part of the incoming client request. The system then checks to see if an entry for the user exists in the LDAP database. This scenario is a good choice for a company that acts as its own Certificate Authority, where the company is assured that if the certificate is verified, then the user is authorized.

- **Certificate Map**
  If you create an object and class that map certificates to users in the LDAP database, you can then configure the system to search for a certificate in the map, and retrieve a user from that map. The system then checks to ensure that the user in the LDAP database is a valid user.

- **Certificate**
  Many LDAP server environments already incorporate certificates into the user information stored in the LDAP database. One way of configuring authorization in LDAP server environments is to configure the system to compare an incoming certificate to the certificate stored in the LDAP database for the user associated with the client request. If the certificate is found in the user’s LDAP profile, access is granted to the user, and the request is granted.

Regardless of the type of certificate-based authorization being performed, the process yields the results shown in Table 10.8.

<table>
<thead>
<tr>
<th>Result of search</th>
<th>Authorization status</th>
</tr>
</thead>
<tbody>
<tr>
<td>No records match</td>
<td>Authorization fails</td>
</tr>
<tr>
<td>One record matches</td>
<td>Authorization succeeds and is subject to groups and roles</td>
</tr>
<tr>
<td>Two or more records match</td>
<td>Authorization fails, due to invalid database entries</td>
</tr>
</tbody>
</table>

*Table 10.8 Search results and corresponding authorization status*
Using groups and roles for LDAP authorization

In addition to enabling certificate-based authorization, you can also configure authorization based on groups and roles.

- **Groups**
  Because LDAP servers already have the concept and structure of groups built into them, the BIG-IP system can include groups in its authorization feature. To enable the use of groups for authorization purposes, you must indicate the base and scope under which the system will search for groups in the LDAP database. Also, you must specify setting values for a group name and a member name. Once you have completed these tasks, the system can search through the list of valid groups until a group is found that has the current user as a member. For information on assigning group-wide privileges for BIG-IP user accounts, see the guide titled *BIG-IP® Local Traffic Manager: Implementations*.

- **Roles**
  Unlike a group, a role is a setting directly associated with a user. Any role-based authorization that the BIG-IP system performs depends on the LDAP database having the concept of roles built into it. To determine if a user should be granted access to a resource, the BIG-IP system searches through the roles assigned to the user and attempts to match that role to a valid role defined by the administrator. For information on assigning group-wide privileges for BIG-IP user accounts, see the guide titled *BIG-IP® Local Traffic Manager: Implementations*.

To implement SSL client certificate LDAP authorization, you must configure the BIG-IP system to access data on a remote LDAP server. To do this, you must create:

- An SSL client certificate LDAP configuration object
- An SSL client certificate LDAP profile

**Important**

When you create SSL Client Certificate LDAP objects and profiles, the BIG-IP system places the them into your current administrative partition. The default profile always resides in partition **Common**. For information on partitions, see the *TMOSTM Management Guide for BIG-IP® Systems*.

After you create these objects, you must then assign the SSL client certificate LDAP profile to a virtual server.

**Note**

If the Create button is unavailable in the Configuration utility when you attempt to create SSL Client Certificate LDAP authentication objects, this indicates that your user role does not grant you permission to create the object.
Creating an SSL client certificate LDAP configuration object

The SSL client certificate LDAP configuration object consists of a set of data and instructions that the corresponding external LDAP server needs when servicing an authorization request from the BIG-IP system.

When you create an SSL client certificate LDAP configuration object, you configure a variety of settings. Table 10.9 lists and describes the settings that you can specify in this configuration object. Note that this table groups the settings into the same categories that you see on the New Authentication Configuration screen. For the detailed procedure on how to configure this object, see To create an SSL client certificate LDAP configuration object, on page 10-26.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to SSL Client Certificate LDAP.</td>
<td>No default value</td>
</tr>
<tr>
<td>Hosts</td>
<td>Lists the IP addresses, including port numbers, of the LDAP servers that the BIG-IP system will use to obtain authorization data.</td>
<td>No default value</td>
</tr>
<tr>
<td>Search Type</td>
<td>Specifies the certificate-based authorization method that the system uses when searching the LDAP database (described in Using SSL certificates for LDAP authorization, on page 10-22). Possible values are User, Certificate Map, and Certificate.</td>
<td>User</td>
</tr>
<tr>
<td>User Base DN</td>
<td>Specifies the search base for the subtree used by the User and Certificate search types. A typical search base is: ou=people,dc=company,dc=com. This setting is required. For more information, see the Search Type setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>User Key</td>
<td>Specifies the name of the attribute in the LDAP database that specifies a user ID. Used by the User and Certificate search type. A typical example of a user key value is uid. This setting is required. For more information, see the Search Type setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Object Class</td>
<td>Specifies a user authentication method. This setting only appears when you set the search type to Certificate.</td>
<td>StrongAuthenticationUser</td>
</tr>
</tbody>
</table>

*Table 10.9 Settings of an SSL client certificate LDAP configuration object*
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate Map Base DN</td>
<td>Specifies the search base for the subtree used by the Certificate Map search types. A typical search base is: ou=people,dc=company,dc=com. This setting is required. For more information, see the Search Type setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Certificate Map Key</td>
<td>Specifies the name of the attribute in the LDAP database that specifies a user ID. Used by the Certificate Map search type. A typical example of a user key value is uid. This setting is required. For more information, see the Search Type setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Use Serial Certificate Map</td>
<td>Causes the system to use the serial number of the client certificate instead of the subject, when you the select Certificate Map search type.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Cache Size</td>
<td>Specifies the maximum size allowed for the SSL session cache. Setting this value to 0 disallows SSL session caching.</td>
<td>20000</td>
</tr>
<tr>
<td>Cache Timeout</td>
<td>Specifies the number of usable lifetime seconds of negotiable SSL session IDs. If this time has expired, a client must negotiate a new session.</td>
<td>300</td>
</tr>
<tr>
<td>Secure</td>
<td>Instructs the BIG-IP system to use secure communication (that is, SSL/TLS) between the system and the LDAP server.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Admin DN</td>
<td>Specifies the distinguished name of an account to which to bind, in order to perform searches. This search account is a read-only account used to do searches. The admin account can be used as the search account. If no admin DN is specified, then no bind is attempted. This setting is required only when a site does not allow anonymous searches.</td>
<td>No default value</td>
</tr>
<tr>
<td>Admin Password</td>
<td>Specifies the password for the search account created on the LDAP server.</td>
<td>No default value</td>
</tr>
<tr>
<td>Confirm Admin Password</td>
<td>Confirms the password specified for the search account created on the LDAP server.</td>
<td>No default value</td>
</tr>
<tr>
<td>Group Base DN</td>
<td>Specifies the search base for the subtree used by group searches. This parameter is only used when specifying valid groups. A typical search base would be similar to the following: ou=people,dc=company,dc=com.</td>
<td>No default value</td>
</tr>
<tr>
<td>Group Key</td>
<td>Indicates the name of the attribute in the LDAP database that specifies the group name in the group subtree.</td>
<td>No default value</td>
</tr>
<tr>
<td>Group Member Key</td>
<td>Indicates the name of the attribute in the LDAP database that specifies members (DNs) of a group.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

*Table 10.9: Settings of an SSL client certificate LDAP configuration object*
To create an SSL client certificate LDAP configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Configurations.
3. In the upper right corner of the screen, click Create. This displays the New Configuration screen.
4. For the Name setting, specify a unique name for the configuration object.
   Note: The name must consist of lowercase characters only.
5. For the Type setting, select SSL Client Certificate LDAP.
6. Modify or retain values for all other settings. (To configure advanced settings, locate the Configuration heading and select Advanced.)
7. Click Finished.
Creating an SSL client certificate LDAP authorization profile

Once you have created an SSL client certificate LDAP configuration object, you must create or configure a corresponding profile. You do this by modifying the default `ssl_cc_ldap` profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object.

In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create an SSL client certificate LDAP profile, you configure a variety of settings. Table 10.10 shows the settings and values that make up an SSL client certificate LDAP profile. For the detailed procedure on creating this type of profile, see To modify the default SSL client certificate LDAP profile, following, or To create a custom SSL client certificate LDAP profile, on page 10-28.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to LDAP (SSL Client Certificate).</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which you want to inherit values.</td>
<td><code>ssl_cc_ldap</code></td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies whether the profile is enabled or disabled. Possible settings are Auto, Enabled, and Disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Configuration</td>
<td>Specifies an existing SSL client certificate LDAP configuration object.</td>
<td>No default value</td>
</tr>
<tr>
<td>Rule</td>
<td>Specifies the name of an existing authentication iRule. If you do not specify an iRule, the BIG-IP system uses the corresponding default iRule.</td>
<td><code>auth_ssl_cc_ldap</code></td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the duration in seconds that an authorization request is idle before timing out. You can use the default value, specify a different value, or select Indefinite. For general information on timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 10.10 Settings of an SSL client certificate LDAP profile
To modify the default SSL client certificate LDAP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Profiles. This displays the list of default authentication profiles.
3. In the Name column, click ssl_cc_ldap.
4. For the Mode setting, select Enabled or Auto.
5. For the Configuration setting, select a configuration object from the list. Note that None is not an allowed setting.
6. For the Rule setting, specify an authentication iRule:
   • If you want to use the default iRule auth_ssl_cc_ldap, leave the setting as is.
   • If you do not want to use the default iRule auth_ssl_cc_ldap, select the name of an existing iRule that you have created.
7. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.
8. Click Finished.

To create a custom SSL client certificate LDAP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Profiles.
3. In the upper right corner of the screen, click Create. This displays the New Profile screen.
4. For the Name setting, specify a unique name for the profile. Note: The name must consist of lowercase characters only.
5. For the Type setting, select SSL Client Certificate LDAP.
6. For the Parent Profile setting, specify a type of profile:
   • If you want to use the default profile ssl_cc_ldap for the parent profile, leave the setting as is.
   • If you want to use a custom profile for the parent profile, select a custom profile name from the list.
7. For the Mode setting, click the Custom box on the right side of the screen, and select Enabled or Auto.
8. For the Configuration setting, click the Custom box on the right side of the screen, and select a configuration object from the list.
9. For the Rule setting, click the Custom box on the right side of the screen, and specify an authentication iRule:
   - If you want to use the default iRule auth_ssl_cc_ldap, leave the setting as is.
   - If you do not want to use the default iRule auth_ssl_cc_ldap, select the name of an existing iRule that you have created.

10. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.

11. Click Finished.

After you have created an SSL Client Certificate LDAP configuration object and an SSL Client Certificate LDAP profile, you must assign the profile to the virtual server by configuring the virtual server’s Authentication Profile setting.
Implementing an SSL OCSP authentication module

An SSL OCSP authentication module is a mechanism for authenticating client connections passing through a BIG-IP system. More specifically, an SSL OCSP authentication module checks the revocation status of an SSL certificate, as part of authenticating that certificate.

**Online Certificate Status Protocol (OCSP)** is a third-party software application and industry-standard protocol that offers an alternative to a certificate revocation list (CRL) when using public-key technology. A CRL is a list of revoked client certificates, which a server system can check during the process of verifying a client certificate.

You implement an SSL OCSP authentication module when you want to use OCSP instead of a CRL as the mechanism for checking the revocation status of SSL certificates.

The BIG-IP system supports both CRLs and the OCSP protocol. If you want to use CRLs instead of OCSP, you configure an SSL profile. For more information on CRLs, see Chapter 9, *Managing SSL Traffic*. For more information on OCSP, see RFC 2560 at URL [http://www.ietf.org](http://www.ietf.org).

Understanding OCSP

Using OCSP to check on the revocation status of client certificates offers distinct advantages over the use of a CRL. The following sections describe the differences between CRLs and OCSP, as well as the way that OCSP operates.

**The limitations of CRLs**

When presented with a client certificate, the BIG-IP system sometimes needs to assess the revocation state of that certificate before accepting the certificate and forwarding the connection to a target server. The standard method of assessing revocation status is a CRL, which is stored in a separate CRL file on each machine in your configuration. Although CRLs are considered to be a standard way of checking revocation status of SSL certificates, a CRL is updated only at fixed intervals, thus presenting a risk that the information in the CRL is outdated at the time that the status check occurs.

Also, having to store a separate CRL file on each machine presents other limitations:

- All CRL files must be kept in sync.
- Having a separate CRL file on each machine poses a security risk.
- Multiple CRL files cannot be administered from a central location.
The benefits of OCSP

OCSP ensures that the BIG-IP system always obtains real-time revocation status during the certificate verification process.

OCSP is based on a client/server model, where a client system requests revocation status of a certificate, and a server system sends the response. Thus, when you implement the SSL OCSP authentication module, the BIG-IP system acts as the OCSP client, and an external system, known as an OCSP responder, acts as the OCSP server. An OCSP responder is therefore an external server that sends certificate revocation status, upon request, to the BIG-IP system.

How does OCSP work?

In general, after receiving an SSL certificate from a client application, the BIG-IP system (acting as an OCSP client) requests certificate revocation status from an OCSP responder, and then blocks the connection until it receives status from that responder. If the status from the responder shows that the certificate has been revoked, the BIG-IP system rejects the certificate and denies the connection. If the status from the responder shows that the certificate is still valid, the BIG-IP system continues with its normal certificate verification process to authenticate the client application.

More specifically, when an application client sends a certificate for authentication, the BIG-IP system follows this process:

• First, the BIG-IP system checks that the signer of the certificate is listed in the trusted CAs file.

• If the certificate is listed, the BIG-IP system then checks to see if the certificate has been revoked. Without OCSP, if the CRL option is configured on the BIG-IP system, the BIG-IP system checks revocation status by reading the certificate revocation list (CRL). With OCSP, however, the BIG-IP system bypasses the CRL and prepares to send a revocation status request to the appropriate OCSP responder.

• Next, the BIG-IP system queries the first responder, even if the responder does not match the certificate authority that signed the client certificate. However, if the first responder fails with a connection error, the BIG-IP system queries the next responder.

• Next, the BIG-IP system attempts to match that CA with a CA listed in an SSL OCSP profile.

• If a match exists, the BIG-IP system checks the target URL within the client certificate’s AuthorityInfoAccess (AIA) field (if the field exists), and uses that URL to send the request for certificate revocation status to the OCSP responder.

• If the Ignore AIA parameter is enabled within the SSL OCSP profile, then the BIG-IP system instead uses the URL specified in the url parameter of the matching SSL OCSP profile to send the request for certificate revocation status.
If no match exists, the BIG-IP system checks the callist setting of another SSL OCSP profile defined on the system. If all SSL OCSP profiles are checked and no match is found, the certificate verification fails, and the BIG-IP system denies the original client request.

Once the BIG-IP system has received certificate revocation status from a responder, the BIG-IP system, when configured to do so, inserts a certificate status header into the original client request. The name of the certificate status header is SSLClientCertificateStatus. For more information on this header, see Prerequisite BIG-IP system profiles, following.

To implement the SSL OCSP authentication module, you must configure the BIG-IP system to access data on a remote OCSP server. To do this, you must create:

- An SSL OCSP responder object
- An SSL OCSP configuration object
- An SSL OCSP profile

**Important**

When you create SSL OCSP objects and profiles, the BIG-IP system places them into your current administrative partition. The default profile always resides in partition Common. For information on partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

After you create these objects, you must then assign the OCSP profile to a virtual server.

A single SSL OCSP profile can target a specific responder, or multiple SSL OCSP profiles can target the same responder. Each responder itself is associated with a certificate authority (CA), and multiple responders can be associated with the same CA.

**Note**

The BIG-IP system allows you to enable both the CRL and the OCSP options. Most users need to enable either one or the other, but not both. However, in the rare case that you want to enable both options, be aware that both the search of the CRL file, and the connection to the responder must be successful. Otherwise, the BIG-IP system cannot obtain status.

**Prerequisite BIG-IP system profiles**

Configuring an SSL OCSP authentication module not only requires the creation of an OCSP responder object and SSL OCSP profile, but also the creation of two other profiles: HTTP and Client SSL.

When you create a Client SSL profile, we recommend that you configure the profile to request, but not require, certificates. This optimizes the BIG-IP system’s use of the SSLClientCertificateStatus header. Note that the
BIG-IP system only inserts this header when previously configured to insert headers into client requests (either through an HTTP profile or through an iRule).

The following sections describe how to create an OCSP responder object, an SSL OCSP configuration object, and an SSL OCSP profile.

◆ Note

*If the Create button is unavailable in the Configuration utility when you attempt to create SSL OCSP authentication objects, this indicates that your user role does not grant you permission to create the object.*

Creating an OCSP responder object

An OCSP responder object is an object that you create that includes a URL for an external OCSP responder. You must create a separate OCSP responder object for each external OCSP responder.

When you subsequently create an OCSP configuration object, the configuration object contains a reference to any OCSP responder objects that you have created.

When you create an OCSP responder object, you configure some settings. Table 10.11 shows the settings and values that make up an SSL OCSP responder object. For the detailed procedure on how to create this object, see *To create an OCSP responder object*, on page 10-35.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>URL</td>
<td>Specifies the URL used to contact the OCSP service on the responder.</td>
<td>No default value</td>
</tr>
<tr>
<td>Certificate Authority File</td>
<td>Specifies the name of the file containing trusted CA certificates used to verify the signature on the OCSP response.</td>
<td>No default value</td>
</tr>
<tr>
<td>Certificate Authority Path</td>
<td>Specifies the name of the path containing trusted CA certificates used to verify the signature on the OCSP response.</td>
<td>No default value</td>
</tr>
<tr>
<td>Verify Other</td>
<td>Specifies the name of the file used to search for an OCSP response signing certificate when the certificate has been omitted from the response.</td>
<td>No default value</td>
</tr>
<tr>
<td>Trust Other</td>
<td>Instructs the BIG-IP system to trust the certificates specified with the Verify Other setting.</td>
<td>Disable</td>
</tr>
</tbody>
</table>

*Table 10.11 Settings of an OCSP responder*
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VA File</td>
<td>Specifies the name of the file containing explicitly-trusted responder certificates. This parameter is needed in the event that the responder is not covered by the certificates already loaded into the responder’s CA store.</td>
<td>No default value</td>
</tr>
<tr>
<td>Signer</td>
<td>Specifies the certificate used to sign an OCSP request. Special meanings: If the certificate is specified but the key is not specified, then the private key is read from the same file as the certificate. If neither the certificate nor the key is specified, then the request is not signed. If the certificate is not specified and the key is specified, then the configuration is considered to be invalid.</td>
<td>No default value</td>
</tr>
<tr>
<td>Signing Key</td>
<td>Specifies a key used to sign an OCSP request.</td>
<td>No default value</td>
</tr>
<tr>
<td>Sign Other</td>
<td>Lists additional certificates to add to an OCSP request.</td>
<td>No default value</td>
</tr>
<tr>
<td>Sign Digest</td>
<td>Specifies the algorithm for signing the request, using the signing certificate and key. Special meanings: This parameter has no meaning if request signing is not in effect (that is, both the request signing certificate and request signing key parameters are empty). This parameter is required only when request signing is in effect.</td>
<td>Sha1</td>
</tr>
<tr>
<td>Validity Period</td>
<td>Specifies the number of seconds that the BIG-IP system should use to specify an acceptable error range. This setting is used when the OCSP responder clock and a client clock are not synchronized, which could cause a certificate status check to fail. This value must be a positive number. This parameter is required.</td>
<td>300</td>
</tr>
<tr>
<td>Status Age</td>
<td>Specifies a time in seconds to compare to the notBefore field of a status response. Used when the status response does not include the notAfter field.</td>
<td>0</td>
</tr>
<tr>
<td>Ignore AIA</td>
<td>Instructs the BIG-IP system to ignore the URL contained in the certificate’s AIA fields, and to always use the URL that the responder specifies instead.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Trust Other</td>
<td>Specifies that the certificates should be explicitly trusted and no other checks should be performed on them.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Allow Certificates</td>
<td>Allows the addition of certificates to an OCSP request.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Verify</td>
<td>Causes the BIG-IP system to verify an OCSP response signature or the nonce values. Used for debugging purposes only.</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

*Table 10.11 Settings of an OCSP responder*
To create an OCSP responder object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
2. From the Authentication menu, choose OCSP Responders.
3. In the upper right corner of the screen, click Create.
4. For the Name setting, type a unique name for the responder object, such as my_ocsp_responder.
   
   Note: The name must consist of lowercase characters only.
5. For the URL setting, type a URL for the external responder.
6. Modify or retain values for all other settings. (To configure advanced settings, locate the Configuration heading and select Advanced).
7. Click Finished.

Table 10.11 Settings of an OCSP responder

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intern</td>
<td>Causes the BIG-IP system to ignore certificates contained in an OCSP response when searching for the signer’s certificate. To use this setting, the signer’s certificate must be specified with either the Verify Other or VA File setting.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Verify Signature</td>
<td>Causes the BIG-IP system to check the signature on the OCSP response. Used for testing purposes only.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Verify Certificate</td>
<td>Causes the BIG-IP system to verify the certificate in the OCSP response.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Certificate Chain</td>
<td>Causes the BIG-IP system to construct a chain from certificates in the OCSP response.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Check Certificates</td>
<td>Causes the BIG-IP system to make additional checks to see if the signer’s certificate is authorized to provide the necessary status information. Used for testing purposes only</td>
<td>Enabled</td>
</tr>
<tr>
<td>Explicit OCSP</td>
<td>Causes the BIG-IP system to explicitly trust that the OCSP response signer’s certificate is authorized for OCSP response signing. If the signer’s certificate does not contain the OCSP signing extension, specification of this setting causes a response to be untrusted.</td>
<td>Enabled</td>
</tr>
</tbody>
</table>
Creating an SSL OCSP configuration object

The SSL OCSP configuration object is an object that references one or more OCSP responder objects.

When you create an OCSP configuration object, you configure a variety of settings. Table 10.12 lists and describes the settings that you can specify in an SSL OCSP configuration object. For the detailed procedure on how to configure this object, see To create an SSL OCSP configuration object, following.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to SSL OCSP.</td>
<td>No default value</td>
</tr>
<tr>
<td>Responders</td>
<td>Specifies the OCSP responders that you want to use for checking that revocation status of SSL certificates.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

Table 10.12 Settings of an SSL OCSP configuration object

To create an SSL OCSP configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Configurations.
3. In the upper right corner of the screen, click Create. This displays the New Configuration screen.
4. For the Name setting, specify a unique name for the configuration object, such as my_ocsp_config. 
   Note: The name must consist of lowercase characters only.
5. For the Type setting, select SSL OCSP.
6. For the Responders setting, specify all responder objects.
7. Click Finished.
Creating an SSL OCSP profile

Once you have created an SSL OCSP configuration object, you must create or configure an SSL OCSP profile. You do this by modifying the default ssl_ocsp profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object.

In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create an OCSP profile, you configure a variety of settings. Table 10.13 shows the settings and values that make up an SSL OCSP profile. For the detailed procedure on creating an OCSP profile, see To modify the default SSL OCSP profile, on page 10-38, or To create a custom SSL OCSP profile, on page 10-38.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to OCSP.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which you want to inherit values.</td>
<td>ssl_ocsp</td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies whether the profile is enabled or disabled. Possible settings are Auto, Enabled, and Disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Configuration</td>
<td>Specifies the name of an existing OCSP configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Rule</td>
<td>Specifies the name of an existing authentication iRule. If you do not specify an iRule, the BIG-IP system uses the corresponding default iRule.</td>
<td>auth_ssl_ocsp</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the duration in seconds that an authentication request is idle before timing out. You can use the default value, specify a different value, or select Indefinite. For general information on timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 10.13  Settings of an SSL OCSP profile
To modify the default SSL OCSP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.
2. From the Authentication menu, choose Profiles.
   This displays the list of default authentication profiles.
3. In the Name column, click ssl_ocsp.
4. For the Mode setting, select Enabled or Auto.
5. For the Configuration setting, select a configuration object from the list. Note that None is not an allowed setting.
6. For the Rule setting, specify an authentication iRule:
   • If you want to use the default iRule auth_ocsp, leave the setting as is.
   • If you do not want to use the default iRule auth_ocsp, select the name of an existing iRule that you have created.
7. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.
8. Click Finished.

To create a custom SSL OCSP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.
2. From the Authentication menu, choose Profiles.
3. In the upper right corner of the screen, click Create.
   This displays the New Profile screen.
4. For the Name setting, specify a unique name for the profile.
   Note: The name must consist of lowercase characters only.
5. For the Type setting, select SSL OCSP.
6. For the Parent Profile setting, specify a type of profile:
   • If you want to use the default profile ssl_ocsp for the parent profile, leave the setting as is.
   • If you want to use a custom profile for the parent profile, select a custom profile name from the list.
7. For the Mode setting, click the Custom box on the right side of the screen, and select Enabled or Auto.
8. For the Configuration setting, click the Custom box on the right side of the screen, and select a configuration object from the list.
9. For the Rule setting, specify an authentication iRule:
   • If you want to use the default iRule auth_ocsp, leave the setting as is.
   • If you do not want to use the default iRule auth_ocsp, select the name of an existing iRule that you have created.

10. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.

11. Click Finished.

After you have created an SSL OCSP responder object, an SSL OCSP configuration object, and an SSL OCSP profile, you must assign the profile to the virtual server by configuring the virtual server’s Authentication Profile setting.

For information on how to configure virtual server settings, see Chapter 2, Configuring Virtual Servers.
Implementing a CRLDP authentication module

A CRLDP authentication module is a mechanism for authenticating client connections passing through a BIG-IP system. This CRLDP authentication feature is based on a technology known as *Certificate Revocation List Distribution Points (CRLDP)*. CRLDP is an industry-standard protocol that offers an alternative method for checking a standard certificate revocation list (CRL) to determine revocation status. CRLDP is also an alternative to using Online Certificate Status Protocol (OCSP). For information on OCSP, see *Implementing an SSL OCSP authentication module*, on page 10-30.

A CRLDP authentication module uses CRL distribution points to check the revocation status of an SSL certificate, as part of authenticating that certificate. *CRL distribution points* are a mechanism used to distribute certificate revocation information across a network. More specifically, a distribution point is a Uniform Resource Identifier (URI) or directory name specified in a certificate that identifies how the server obtains CRL information. Distribution points can be used in conjunction with CRLs to configure certificate authorization using any number of LDAP servers.

You implement a CRLDP authentication module when you want the BIG-IP system to use CRL distribution points as the mechanism for checking the revocation status of SSL certificates.

To implement a CRLDP authentication module, you must configure the BIG-IP system to access data on a remote CRLDP server. To do this, you must create:

- One or more high-level CRLDP server objects
- A CRLDP configuration object
- A CRLDP profile

**Important**

When you create CRLDP objects and profiles, the BIG-IP system places them into your current administrative partition. The default profile always resides in partition *Common*. For information on partitions, see the *TMOSTM Management Guide for BIG-IP® Systems*.

After you create these objects, you must then assign the CRLDP profile to a virtual server.

**Note**

If the *Create* button is unavailable in the Configuration utility when you attempt to create CRLDP authentication objects, this indicates that your user role does not grant you permission to create the object.
Prerequisite task

Before you configure the CRLDP authentication module, you must create a Client SSL profile. When you create the Client SSL profile, you must set its Client Certificate setting to Request or Require. For information on creating a Client SSL profile, see Chapter 9, Managing SSL Traffic.

Once you have created the Client SSL profile and configured the Client Certificate setting correctly, you can create a CRLDP server object, an CRLDP configuration object, and a CRLDP profile.

Creating a CRLDP server object

When you create a CRLDP server object, you configure some settings. Table 10.14 shows the settings and values that make up a default CRLDP server object. For the detailed procedure on creating a server object, see To create a CRLDP server object, following.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the CRLDP server object, such as my_crldp_server. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Address</td>
<td>Specifies an IP address for the CRLDP server. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Service Port</td>
<td>Specifies the port for CRLDP authentication traffic.</td>
<td>389</td>
</tr>
<tr>
<td>Base DN</td>
<td>Specifies the LDAP base distinguished name for certificates that specify the CRL distribution point in directory name (dirName) format. Used when the value of the X509v3 attribute crlDistributionPoints is of type dirName. In this case, the BIG-IP system attempts to match the value of the crlDistributionPoints attribute to the Base DN value. An example of a Base DN value is cn=lxxx,dc=f5,dc=com.</td>
<td>No default value</td>
</tr>
<tr>
<td>Reverse DN</td>
<td>Specifies in which order the system is to attempt to match the Base DN value to the value of the X509v3 attribute crlDistributionPoints. Possible values are Enabled and Disabled. When set to Enabled, the system matches the base DN from left to right, or from the beginning of the DN string, to accommodate dirName strings in certificates such as c=us,st=wa,l=sea,ou=f5,cn=xxx.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Table 10.14 Settings of a CRLDP sever definition
To create a CRLDP server object

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**. The Profiles screen opens.
2. From the Authentication menu, choose CRLDP Servers. This displays the CRLDP Server List screen.
3. In the upper right corner of the screen, click **Create**.
4. For the **Name** setting, type a unique name for the CRLDP server object, such as `my_crldp_server`.
   \*Note: The name must consist of lowercase characters only.*
5. For the **Address** setting, type an IP address for the remote CRLDP server.
6. For the **Base DN** setting, type a base Distinguished Name.
7. From the **Reverse DN** list, retain the default value or select **Enabled**.
8. Click **Finished**.
9. For redundant CRLDP servers, repeat these steps to create additional server objects.

Creating a CRLDP configuration object

When you create a CRLDP configuration object, you configure a variety of settings. Table 10.15 shows the settings and values that make up a CRLDP configuration object. Note that this table groups the settings into the same categories that you see on the New Authentication Configuration screen. For the detailed procedure on how to create this configuration object, see *To create a CRLDP configuration object*, on page 10-43.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to <strong>CRLDP</strong>.</td>
<td>No default value</td>
</tr>
<tr>
<td>Connection Timeout</td>
<td>Specifies the number of seconds before the connection times out.</td>
<td>15</td>
</tr>
</tbody>
</table>

*Table 10.15 Settings of a CRLDP configuration object*
To create a CRLDP configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
The Profiles screen opens.

2. From the Authentication menu, choose Configurations.

3. In the upper right corner of the screen, click Create.
   This displays the New Configuration screen.

4. For the Name setting, specify a unique name for the configuration object, such as my_crldp_config.
   Note: The name must consist of lowercase characters only.

5. For the Type setting, select CRLDP.
   The screen expands to show several settings.

6. Modify or retain values for all settings shown.

7. Click Finished.

Creating a CRLDP profile

Once you have created a CRLDP configuration object, you must create or configure a CRLDP profile. You do this by modifying the default ssl_crldp profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object.
In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create a CRLDP profile, you configure a variety of settings. Table 10.16 shows the settings and values that make up a CRLDP profile. For the detailed procedure on creating a CRLDP profile, see To create a custom CRLDP profile, on page 10-45.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to CRLDP.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which you want to inherit values.</td>
<td>ssl_crldp</td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies whether the profile is enabled or disabled. Possible settings are Enabled and Disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Configuration</td>
<td>Specifies an existing CRLDP configuration object.</td>
<td>No default value</td>
</tr>
<tr>
<td>Rule</td>
<td>Specifies the name of an existing authentication rule. If you do not specify an iRule, the BIG-IP system uses the corresponding default iRule.</td>
<td>auth_ssl_crldp</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>Specifies the duration in seconds that an authentication or authorization request is idle before timing out. You can use the default value, specify a different value, or select Indefinite. For general information on timeout values, see Chapter 1, Introducing Local Traffic Management.</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 10.16 Settings of a CRLDP profile

To modify the default CRLDP profile

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles.
   The Profiles screen opens.
2. From the Authentication menu, choose Profiles.
   This displays the list of default authentication profiles.
3. In the Name column, click ssl_crldp.
4. For the Mode setting, verify that the mode is set to Enabled.
5. For the Configuration setting, click the Custom box on the right side of the screen, and select a configuration object from the list. Note that None is not an allowed setting.
6. For the **Rule** setting, specify an authentication iRule:
   - If you want to use the default iRule `auth_ssl_crldp`, leave the setting as is.
   - If you do not want to use the default iRule `auth_ssl_crldp`, click the Custom box on the right side of the screen, and select the name of an existing iRule that you have created.

7. For the **Idle Timeout** setting, retain the default value, specify a new value, or select **Indefinite**.

8. Click **Finished**.

**To create a custom CRLDP profile**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**. The Profiles screen opens.

2. From the Authentication menu, choose Profiles.

3. In the upper right corner of the screen, click **Create**. This displays the New Profile screen.

4. In the **Name** setting, type a unique name for the CRLDP profile, such as `my_crldp_profile`.  
   
   **Note**: The name must consist of lowercase characters only.

5. In the **Type** setting, select **CRLDP**. The screen expands to show additional settings

6. For the **Parent Profile** setting, specify a type of profile:
   - If you want to use the default profile `ssl_crldp` for the parent profile, leave the setting as is.
   - If you want to use a custom profile for the parent profile, select a custom profile name from the list.

7. In the **Mode** setting, verify that the mode is set to **Enabled**.

8. In the **Configuration** setting, click the Custom box on the right side of the screen, and select a configuration object from the list.

9. For the **Rule** setting, specify an authentication iRule:
   - If you want to use the default iRule `auth_ssl_crldp`, leave the setting as is.
   - If you do not want to use the default iRule `auth_ssl_crldp`, click the Custom box on the right side of the screen, and select the name of an existing iRule that you have created.

10. For the **Idle Timeout** setting, retain the default value, specify a new value, or select **Indefinite**.

11. Click **Finished**.
After you have created a CRLDP server object, a CRLDP configuration object, and a CRLDP profile, you must assign the profile to the virtual server by configuring the virtual server’s **Authentication Profile** setting.

For information on how to configure virtual server settings, see Chapter 2, *Configuring Virtual Servers*. 

Implementing a Kerberos Delegation authentication module

A Kerberos Delegation authentication module is a mechanism for authenticating client connections passing through a BIG-IP system. You can use this module when you are using Microsoft® Windows® Integrated Authentication to authenticate application traffic.

**Important**

Prior to implementing a Kerberos Delegation module, you must add a Domain Name Server (DNS) server to the BIG-IP system, and add the BIG-IP system to the domain. For more information, see the guide titled BIG-IP® Local Traffic Manager: Implementations.

The Kerberos Delegation module obtains delegated Kerberos credentials for the client principal, and then retrieves Kerberos credentials for the server-side principal. The Kerberos Delegation module essentially acts as a proxy for Kerberos credentials. That is, when connecting to a server that is inside its domain, the browser client fetches Kerberos credentials. These credentials, known as delegated credentials, are passed to the BIG-IP system, which in turn retrieves credentials for the real server that is on the back end, and passes those credentials back to the client.

To implement the Kerberos Delegation authentication module, you must configure the BIG-IP system to access data on a remote server. To do this, you must create:

- A Kerberos Delegation configuration object
- A Kerberos Delegation profile

**Important**

When you create a Kerberos Delegation object and profile, the BIG-IP system places them into your current administrative partition. The default profile always resides in partition Common. For information on partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

After you create these objects, you must then assign the Kerberos Delegation profile to a virtual server.

**Note**

If the Create button is unavailable in the Configuration utility when you attempt to create Kerberos Delegation authentication objects, this indicates that your user role does not grant you permission to create the object.
Creating a Kerberos Delegation configuration object

When you create a Kerberos Delegation configuration object, you configure a variety of settings. Table 10.17 shows the settings and values that make up a Kerberos Delegation configuration object. Note that this table groups the settings into the same categories that you see on the New Authentication Configuration screen. For the detailed procedure on how to create this object, see To create a Kerberos Delegation configuration object, following.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the configuration object. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to Kerberos Delegation.</td>
<td>No default value</td>
</tr>
<tr>
<td>Client Principal Name</td>
<td>Specifies the name of the client principal, using the format HTTP/&lt;fqdn&gt;, where fqdn is the fully-qualified domain name of the virtual server you create. For more information, see the guide titled BIG-IP® Local Traffic Manager: Implementations.</td>
<td>No default value</td>
</tr>
<tr>
<td>Server Principal Name</td>
<td>Specifies the name of the principal of the back-end web server, using the format HTTP/&lt;fqdn&gt;, where fqdn is the fully-qualified domain name of the web server in the pool. For more information, see the guide titled BIG-IP® Local Traffic Manager: Implementations.</td>
<td>No default value</td>
</tr>
<tr>
<td>Debug Logging</td>
<td>Enables SYSLOG debugging information at LOG_DEBUG level. Not recommended for normal use.</td>
<td>Disable</td>
</tr>
</tbody>
</table>

Table 10.17 Settings of a Kerberos Delegation configuration object

To create a Kerberos Delegation configuration object

1. On the Main tab of the navigation pane, expand Local Traffic, and click Profiles. The Profiles screen opens.
2. From the Authentication menu, choose Configurations.
3. In the upper right corner of the screen, click Create. This displays the New Configuration screen.
4. For the Name setting, specify a unique name for the configuration object, such as my_kerberos_config.
   Note: The name must consist of lowercase characters only.
5. For the Type setting, select Kerberos Delegation. The screen expands to show several settings.
6. Modify or retain values for all settings shown.
   (To configure advanced settings, locate the Configuration heading and select Advanced.)

7. Click Finished.

Creating a Kerberos Delegation profile

Once you have created a Kerberos Delegation configuration object, you must create or configure a Kerberos Delegation profile. You do this by modifying the default krbdelegate profile or by creating a custom profile that inherits the default profile settings. An important function of the authentication profile is to reference an existing configuration object.

In most cases, the default profile should suit your needs. However, even if you use the default profile, you must still modify it to specify the corresponding configuration object that you created.

If you choose to create a custom profile, you must specify a parent profile (either a custom profile or the default profile) that contains the values that you want the new profile to inherit.

When you create a Kerberos Delegation profile, you configure a variety of settings. Table 10.18 shows the settings and values that make up a Kerberos Delegation profile. For the detailed procedure on creating a Kerberos Delegation profile, see To modify the default Kerberos Delegation profile, following, or To create a custom Kerberos Delegation profile, on page 10-50.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the profile. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of authentication module you want to implement. You must set this value to Kerberos Delegation.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>Specifies the profile from which you want to inherit values.</td>
<td>krbdelegate</td>
</tr>
<tr>
<td>Mode</td>
<td>Specifies whether the profile is enabled or disabled. Possible settings are Auto, Enabled, and Disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Configuration</td>
<td>Specifies an existing Kerberos Delegation configuration object.</td>
<td>No default value</td>
</tr>
<tr>
<td>Rule</td>
<td>Specifies the name of an existing authentication rule. If you do not specify an iRule, the BIG-IP system uses the corresponding default iRule.</td>
<td>auth_krbdelegate</td>
</tr>
</tbody>
</table>

Table 10.18 Settings of a Kerberos Delegation profile
To modify the default Kerberos Delegation profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The Profiles screen opens.

2. From the Authentication menu, choose Profiles.
   This displays the list of default authentication profiles.

3. In the Name column, click **krbdelegate**.

4. For the **Mode** setting, retain the default value, **Enabled**.

5. For the **Configuration** setting, select a configuration object from the list. Note that **None** is not an allowed setting.

6. For the **Rule** setting, specify an authentication iRule:
   - If you want to use the default iRule **auth_krbdelegate**, leave the setting as is.
   - If you do not want to use the default iRule **auth_krbdelegate**, select the name of an existing iRule that you have created.

7. For the **Idle Timeout** setting, retain the default value, specify a new value, or select **Indefinite**.

8. Click **Finished**.

To create a custom Kerberos Delegation profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The Profiles screen opens.

2. From the Authentication menu, choose Profiles.
3. In the upper right corner of the screen, click Create. This displays the New Profile screen.

4. For the Name setting, specify a unique name for the profile.
   
   Note: The name must consist of lowercase characters only.

5. For the Type setting, select Kerberos Delegation.

6. For the Parent Profile setting, specify a type of profile:
   
   - If you want to use the default profile krbdelegate for the parent profile, leave the setting as is.
   - If you want to use a custom profile for the parent profile, select a custom profile name from the list.

7. For the Mode setting, retain the default value, Enabled.

8. For the Configuration setting, select a configuration object from the list.

9. For the Rule setting, specify an authentication iRule:
   
   - If you want to use the default iRule auth_krbdelegate leave the setting as is.
   - If you do not want to use the default iRule auth_krbdelegate, select the name of an existing iRule that you have created.

10. For the Idle Timeout setting, retain the default value, specify a new value, or select Indefinite.

11. Click Finished.

After you have created a Kerberos Delegation configuration object and a Kerberos Delegation profile, you must assign the profile to the virtual server by configuring the virtual server’s Authentication Profile setting.

For information on how to configure virtual server settings, see Chapter 2, Configuring Virtual Servers.
Using Additional Profiles

- Introducing other types of profiles
- Configuring a OneConnect profile
- Configuring an NTLM profile
- Configuring a Statistics profile
- Configuring a Stream profile
Introducing other types of profiles

In addition to the profiles described in previous chapters, you can configure these profiles:

- OneConnect
- NTLM
- Statistics
- Stream

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.

The remainder of this chapter lists the traffic-management settings contained in OneConnect™, NTLM, Statistics, and Stream profiles. For information on configuring other types of profiles, see the following chapters:

- For information on the HTTP, FTP, RTSP, SIP, and iSession profiles, see Chapter 6, Managing Application Layer Traffic.
- For information on the profiles for session persistence, see Chapter 7, Enabling Session Persistence.
- For information on the Fast L4, Fast HTTP, HTTP Class, TCP, UDP, and SCTP profiles, see Chapter 8, Managing Protocol Profiles.
- For information on the SSL profiles, see Chapter 9, Managing SSL Traffic.
- For information on the profiles for remote authentication of application traffic, see Chapter 10, Authenticating Application Traffic.
Configuring a OneConnect profile

The OneConnect profile is a configuration tool for enabling connection pooling on a BIG-IP system. **Connection pooling** optimizes the way that the BIG-IP system handles connections. When connection pooling is enabled on a BIG-IP system, client requests can utilize existing, server-side connections, thus reducing the number of server-side connections that a server must open to service those requests.

The BIG-IP system can pool connections from multiple virtual servers if those virtual servers reference the same OneConnect profile and the same pool.

**Important**

To enable connection pooling, you must also enable a related feature known as the OneConnect Transformations feature. You enable this feature from within an HTTP profile. The OneConnect Transformations HTTP profile setting applies to HTTP/1.0 connections, and when enabled, causes the system to transform the value of the **Connection** header in an HTTP request to **Keep-Alive**, to keep the connection open.

As previously described in Chapter 2, Configuring Virtual Servers, the standard address translation mechanism on the BIG-IP system translates only the destination IP address in a request and not the source IP address (that is, the client node’s IP address). However, when the OneConnect feature is enabled, allowing multiple client nodes to re-use a server-side connection, the source IP address in the header of each client node’s request is always the IP address of the specific client node that initiated the server-side connection. Although this does not affect traffic flow, you might see evidence of this when viewing certain types of system output.

Table 11.1 lists and describes the settings of a OneConnect profile type.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This setting specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>oneconnect</td>
</tr>
<tr>
<td>Source Mask</td>
<td>The BIG-IP system applies the value of this setting to the source address to determine its eligibility for reuse. A mask of 0 causes the BIG-IP system to share reused connections across all clients. A host mask (that is, all 1 values in binary), causes the BIG-IP system to share only those reused connections originating from the same client IP address.</td>
<td>0.0.0.0</td>
</tr>
</tbody>
</table>

*Table 11.1 Settings of a OneConnect profile*
Using Additional Profiles

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Size</td>
<td>The setting defines the maximum number of connections that the BIG-IP system holds in the connection reuse pool. If the pool is already full, then a server-side connection closes after the response is completed. Important: When using a OneConnect profile, the value of the Maximum Size setting must be at least one number smaller than the connection limit set for a pool member. This is because the BIG-IP system includes connections in the connection re-use pool as part of a pool member's connection limit. Therefore, if the number of connections for a pool member reaches the maximum number allowed, the virtual server can become unavailable.</td>
<td>10000</td>
</tr>
<tr>
<td>Maximum Age</td>
<td>This setting defines the maximum number of seconds allowed for a connection in the connection re-use pool. For any connection with an age higher than this value, the BIG-IP system removes that connection from the re-use pool.</td>
<td>86400</td>
</tr>
<tr>
<td>Maximum Reuse</td>
<td>This setting specifies the maximum number of times that a server-side connection can be re-used.</td>
<td>1000</td>
</tr>
<tr>
<td>Idle Timeout Override</td>
<td>This setting specifies the number of seconds that a connection is idle before the connection flow is eligible for deletion. You can use this setting to increase the timeout value for connections once they are pooled for re-use. Possible values are Disabled, Indefinite, or Specify (a numeric value that you specify).</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Table 11.1 Settings of a OneConnect profile

For information on iRules™ ONECONNECT commands, see the F5 Networks DevCentral web site http://devcentral.f5.com, and Chapter 17, Writing iRules.
Configuring an NTLM profile

*NT Lan Manager (NTLM)* is an industry-standard technology that uses an encrypted challenge/response protocol to authenticate a user without sending the user's password over the network. Instead, the system requesting authentication performs a calculation to prove that the system has access to the secured NTLM credentials. NTLM credentials are based on data such as the domain name and user name, obtained during the interactive logon process.

The NTLM profile within BIG-IP Local Traffic Manager optimizes network performance when the system is processing NT LAN Manager traffic. When both an NTLM profile and a OneConnect profile are associated with a virtual server, the local traffic management system can take advantage of server-side connection pooling for NTLM connections.

Table 11.2 lists and describes the settings of an NTLM profile type.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This setting specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>ntlm</td>
</tr>
<tr>
<td>Insert Cookie Name</td>
<td>Specifies a cookie name that the system inserts in the cookie.</td>
<td>NTLMconnpool</td>
</tr>
<tr>
<td>Insert Cookie Passphrase</td>
<td>Specifies a cookie passphrase that the system inserts in the cookie. The default is blank, indicating that no passphrase is inserted.</td>
<td>No default value</td>
</tr>
<tr>
<td>Key By Existing Cookie</td>
<td>Specifies whether the system uses the existing cookie as the key.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Cookie Name</td>
<td>Specifies the name of the cookie.</td>
<td>mycookie</td>
</tr>
<tr>
<td>Key By NTLM Domain</td>
<td>Specifies whether the system uses the NTLM domain as the key.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Key By Client IP Address</td>
<td>Specifies whether the system uses the client IP address as the key.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Key By NTLM Target</td>
<td>Specifies whether the system uses the NTLM target as the key.</td>
<td>Disabled (Unchecked)</td>
</tr>
<tr>
<td>Key by NTLM User</td>
<td>Specifies whether the system uses the NTLM user as the key.</td>
<td>Enabled (Checked)</td>
</tr>
<tr>
<td>Key By NTLM Workstation</td>
<td>Specifies whether the system uses the NTLM workstation as the key.</td>
<td>Disabled (Unchecked)</td>
</tr>
</tbody>
</table>

*Table 11.2 Settings of an NTLM profile*
Note

In addition to creating an NTLM profile, you must also create a OneConnect profile. Both profiles must then be associated with the virtual server. For information on creating a OneConnect profile, see Configuring a OneConnect profile, on page 11-2.
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 11.1 shows this configuration.

```tcl
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 11.1 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](http://devcentral.f5.com), and Chapter 17, *Writing iRules*. 
Configuring a Stream profile

You can use the Stream profile to search and replace strings within a data stream, such as a TCP connection. Table 11.3 lists and describes the settings of a Stream profile type.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This setting specifies a unique name for the profile.</td>
<td>No default value</td>
</tr>
<tr>
<td>Parent Profile</td>
<td>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.</td>
<td>stream</td>
</tr>
<tr>
<td>Source</td>
<td>Specifies the source string for which to search.</td>
<td>No default value</td>
</tr>
<tr>
<td>Target</td>
<td>Specifies the target string to replace.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

Table 11.3 Settings of a Stream profile

Note that list types are case-sensitive for pattern strings. For example, the system treats the pattern string `www.f5.com` differently from the pattern string `www.F5.com`. You can override this case sensitivity by using the Linux `regexp` command.

For information on iRules `STREAM` commands, see the F5 Networks DevCentral web site `http://devcentral.f5.com`, and Chapter 17, Writing iRules.
Configuring Monitors

- Introducing monitors
- Creating a custom monitor
- Special configuration considerations
- Associating monitors with pools and nodes
- Managing monitors
Introducing monitors

The BIG-IP® local traffic management system can monitor the health or performance of either pool members or nodes. The local traffic management system supports these methods of monitoring:

- **Simple monitoring**
  
  Simple monitoring merely determines whether the status of a node is **up** or **down**. Simple monitors do not monitor pool members (and therefore, individual protocols, services, or applications on a node), but only the node itself. The system contains two types of simple monitors, ICMP and TCP_ECHO.

- **Active monitoring**
  
  Active monitoring checks the status of a pool member or node on an ongoing basis, at a set interval. If a pool member or node being checked does not respond within a specified timeout period, or the status of a pool member or node indicates that performance is degraded, the BIG-IP system can redirect the traffic to another pool member or node. There are many types of active monitors. Each type of active 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 node. A WMI type of monitor allows you to monitor the performance of a pool, pool member, or node that is running the Windows Management Instrumentation (WMI) software. Active monitors fall into two categories: Extended Content Verification (ECV) monitors and Extended Application Verification (EAV) monitors.

- **Passive monitoring**
  
  Passive monitoring occurs as part of a client request. This kind of monitoring checks the health of a pool member based on a specified number of connection attempts or data request attempts that occur within a specified time period. If, after the specified number of attempts within the defined interval, the system cannot either connect to the server or receive a response, or if the system receives a bad response, the system marks the pool member as **down**. There is only one type of passive monitor, called an Inband monitor.

To assist you in deciding which monitoring method to use, Table 12.1, on page 12-2, shows the benefits and constraints of each monitoring method.
Chapter 12

Summary of monitor types

The local traffic management system includes a wide variety of monitors. You can choose which types of monitors you want to associate with a given pool, pool member, or node.

Table 12.2 lists and describes all monitor types available with the BIG-IP system.

<table>
<thead>
<tr>
<th>Monitor Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>Checks the status of a node, using Internet Control Message Protocol (ICMP).</td>
</tr>
<tr>
<td>TCP Echo</td>
<td>Checks the status of a node, using Transmission Control Protocol (TCP).</td>
</tr>
<tr>
<td>Gateway ICMP</td>
<td>Checks nodes in a pool that implements gateway failsafe for high availability.</td>
</tr>
<tr>
<td>HTTP</td>
<td>Verifies the Hypertext Transfer Protocol (HTTP) service by attempting to receive specific content from a web page.</td>
</tr>
<tr>
<td>HTTPS</td>
<td>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.</td>
</tr>
<tr>
<td>TCP</td>
<td>Verifies the Transmission Control Protocol (TCP) service by attempting to receive specific content from a node.</td>
</tr>
</tbody>
</table>

Table 12.1 Comparison of monitoring methods

<table>
<thead>
<tr>
<th>Monitoring Method</th>
<th>Benefits</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>• Works well when you only need to determine the up or down status of a node.</td>
<td>• Can check the health of a node only, and not a pool member.</td>
</tr>
<tr>
<td>Active</td>
<td>• Can check for specific responses</td>
<td>• Creates additional network traffic beyond the client request and server response</td>
</tr>
<tr>
<td></td>
<td>• Can run with or without client traffic</td>
<td>• Can be slow to mark a pool member as down</td>
</tr>
<tr>
<td>Passive</td>
<td>• Creates no additional network traffic beyond the client request and server response</td>
<td>• Cannot check for specific responses</td>
</tr>
<tr>
<td></td>
<td>• Can mark a pool member as down quickly, as long as there is some amount of network traffic</td>
<td>• Can potentially be slow to mark a pool member as up</td>
</tr>
</tbody>
</table>

Table 12.2 Monitor types available on a BIG-IP system

Important

For a detailed description of each monitor type, see Appendix A, Health and Performance Monitors.
**Monitor Types** | **Description**
---|---
TCP Half Open | Monitors the associated service by sending a TCP SYN packet to the service. As soon as the monitor receives the SYN-ACK packet, the monitor marks the service as **up**.
External | Allows users to monitor services using their own programs.
FirePass | Monitors FirePass systems.
FTP | Verifies the File Transfer Protocol (FTP) service by attempting to download a specific file to the `/var/tmp` directory on an BIG-IP system. Once downloaded successfully, the file is not saved.
IMAP | Verifies the Internet Message Access Protocol (IMAP) by attempting to open a specified mail folder on a server. This monitor is similar to the **pop3** monitor.
LDAP | Verifies the Lightweight Directory Access Protocol (LDAP) service by attempting to authenticate the specified user.
Module Score | Enables Global Traffic Manager and Local Traffic Manager systems to load balance in a proportional manner to Local Traffic Manager virtual servers associated with the Web Accelerator and Application Security Manager modules.
MSSQL | Verifies Microsoft® Windows® SQL-based services. Before using this type of monitor, you must install a set of Microsoft® JDBC Java™ Archive (JAR) files. For information on installing these files, see Appendix B, *Additional Monitor Considerations*.
NNTP | Verifies the Usenet News protocol (NNTP) service by attempting to retrieve a newsgroup identification string from the server.
Oracle | Verifies services based on Oracle® by attempting to perform an Oracle login to a service.
POP3 | Verifies the Post Office Protocol (pop3) service by attempting to connect to a pool, pool member, or node, log on as the specified user, and log off.
RADIUS | Verifies the Remote Access Dial-in User Service (RADIUS) service by attempting to authenticate the specified user.
Real Server | Checks the performance of a pool, pool member, or node that is running the RealServer data collection agent, and then dynamically load balances traffic accordingly.
RPC | Verifies the availability of RPC (Remote Procedure Call) servers using the **rpcinfo** command.
SASP | Uses the Server/Application State Protocol (SASP) to communicate and verify availability of resources managed through the IBM® Group Workload Manager.
SIP | Checks the status of Session Initiation Protocol (SIP) Call-ID services on a device. The SIP protocol enables real-time messaging, voice, data, and video.
SMB | Employs the Server Message Block (SMB) to verify whether either an SMB/CIFS server or a specific share on that server is available.
SMTP | Checks the status of a pool, pool member, or node by issuing standard Simple Mail Transport Protocol (SMTP) commands.

*Table 12.2* Monitor types available on a BIG-IP system
About 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 BIG-IP system assigns default values. For example, Figure 12.1 shows that an ICMP-type monitor has these settings and default values.

<table>
<thead>
<tr>
<th>Monitor Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP DCA</td>
<td>Checks the current CPU, memory, and disk usage of a pool, pool member, or node that is running an SNMP data collection agent, and then dynamically load balances traffic accordingly.</td>
</tr>
<tr>
<td>SNMP DCA Base</td>
<td>Checks the current user usage of a pool, pool member, or node that is running an SNMP data collection agent, and then dynamically load balances traffic accordingly. The way that you configure the monitor settings determines the data that the BIG-IP system collects.</td>
</tr>
<tr>
<td>SOAP</td>
<td>Tests a Web service based on the Simple Object Access Protocol (SOAP).</td>
</tr>
<tr>
<td>UDP</td>
<td>Verifies the User Datagram Protocol (UDP) service by attempting to send UDP packets to a pool, pool member, or node and receiving a reply.</td>
</tr>
<tr>
<td>WMI</td>
<td>Checks the performance of a pool, pool member, or node that is running the Windows Management Infrastructure (WMI) data collection agent and then dynamically load balances traffic accordingly.</td>
</tr>
<tr>
<td>Inband</td>
<td>Determines the availability of a pool member based on a specified number of failed connection attempts or data request attempts within a specified time interval.</td>
</tr>
</tbody>
</table>

Table 12.2 Monitor types available on a BIG-IP system

The settings in Figure 12.1 specify that an ICMP type of monitor is configured to check the status of an IP address every five seconds, and to time out every 16 seconds. The destination IP address that the monitor checks is specified by the Alias Address setting, with the value * All Addresses. Thus, in the preceding example, all IP addresses with which the monitor is associated are checked. For more information on monitor settings, see Special configuration considerations, on page 12-10.
Overview of monitor implementation

You implement monitors using either the Configuration utility or a command line utility. The task of implementing a monitor varies depending on whether you are using a pre-configured monitor or creating a custom monitor. A pre-configured monitor is an existing monitor that the BIG-IP system provides for you, with its settings already configured. A custom monitor is a monitor that you create based on one of the allowed monitor types.

If you want to implement a pre-configured monitor, you need only associate the monitor with a pool, pool member, or node, and then configure the virtual server to reference the relevant pool. If you want to implement a custom monitor, you must first create the custom monitor. Then you can associate the custom monitor with a pool, pool member, or node, and configure the virtual server to reference the pool.

For the procedure on creating a custom monitor, see Creating a custom monitor, on page 12-8. For the procedure on associating a monitor with a pool or pool member, or with a node, see Associating monitors with pools and nodes, on page 12-15.

For information on configuring a virtual server to reference a pool, see Chapter 2, Configuring Virtual Servers.

Using pre-configured monitors

For a subset of monitor types, the BIG-IP system includes a set of pre-configured monitors. 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 a monitor. 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 BIG-IP system includes are:

- gateway_icmp
- http
- https
- https_443
- icmp
- inband
- real_server
- snmp_dca
- tcp
- tcp_echo
An example of a pre-configured monitor is the **icmp** monitor. Figure 12.2 shows the **icmp** monitor, with values configured for its **Interval**, **Timeout**, and **Alias Address** settings. Note that the **Interval** value is 5, the **Timeout** value is 16, the **Transparent** value is No, and the **Alias Address** value is *All Addresses*.

<table>
<thead>
<tr>
<th>Name</th>
<th>icmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>ICMP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
</tbody>
</table>

*Figure 12.2 The icmp pre-configured monitor*

If the **Interval**, **Timeout**, **Transparent**, and **Alias Address** values meet your needs, you simply assign the **icmp** pre-configured monitor directly to a pool, pool member, or node, using the **Pools** or **Nodes** screens within the Configuration utility. In this case, you do not need to use the Monitors screens, unless you simply want to view the values of the pre-configured monitor settings.

**Important**

All pre-configured monitors reside in partition **Common**. For information on partitions, see the *TMOSTM Management Guide for BIG-IP® Systems*.

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

**Using custom monitors**

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 12-2.)

**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.

*Figure 12.3, on page 12-7, shows an example of a custom ICMP-type monitor called **my_icmp**, which is based on the pre-configured monitor **icmp**.*
Note that the Interval value has been changed to 10, and the Timeout value to 20. The other settings retain the values defined in the pre-configured monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>my_icmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>ICMP</td>
</tr>
<tr>
<td>Interval</td>
<td>10</td>
</tr>
<tr>
<td>Timeout</td>
<td>20</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
</tbody>
</table>

Figure 12.3 A custom monitor based on a pre-configured monitor

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 BIG-IP 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 12-8.
Creating a custom monitor

When you create a custom monitor, you use the Configuration utility or a command line 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, simply choose the one that corresponds to the service you want to check. For example, if you want to 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 pool member (for example HTTP and HTTPS), you can associate more than one monitor on that pool or pool member. For more information, see Chapter 4, Configuring Load Balancing Pools.

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 node 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 detailed information on monitor types, see Appendix A, Health and Performance Monitors.

◆ Important

When you create a custom monitor, the BIG-IP system places the monitor into your current administrative partition. For more information on administrative partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

To create a custom monitor

1. On the Main tab of the navigation pane, expand Local Traffic, and click Monitors.
   The Monitors screen opens.
2. In the upper right corner of the screen, click Create.
   The New Monitor screen opens.
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a custom monitor.
3. For the Type setting, select the type of monitor that you want to create.
   If a monitor of that type already exists, Import Settings appears.
4. Specify a name, based on one of these settings:
   • If Import Settings appears, choose a monitor name from the list.
   • If a monitor of the type you selected does not exist, in the Name box, type a unique name for the custom monitor.

5. In the Configuration section of the screen, select Advanced. This allows you to modify additional default settings.

6. Configure all settings shown.

7. Click Finished.

◆ Important

When defining values for custom monitors, make sure you avoid using any values that are on the list of reserved keywords. For more information, see solution number 3653 (for 9.0+ systems) on the Ask F5® technical support web site.
Special configuration considerations

Every pre-configured or custom monitor has settings with some default values assigned. You will find certain topics useful when changing these default values:

- **Setting destinations**
  For more information, see *Setting destinations*, following.

- **Using transparent and reverse modes**
  For more information, see *Using transparent and reverse modes*, on page 12-11.

- **Configuring the Manual Resume setting**
  For more information, see *Configuring the Manual Resume setting*, on page 12-12.

- **Configuring the Check Until Up setting**
  For more information, see *Configuring the Check Until Up setting*, on page 12-14.

Setting destinations

By default, the value for the **Alias Address** setting in the monitors is set to the wildcard *Addresses*, and the **Alias Service Port** setting is set to the wildcard *Ports*. This value causes the monitor instance created for a pool, pool member, or node to take that node’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, pool member, or node.

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 \texttt{get} or \texttt{hurl} in place of \texttt{Send String} and \texttt{Receive String} statements. For FTP monitors specifically, the \texttt{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, pool member, or node by an unspecified route, and to mark the node \textbf{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 \textbf{Transparent} or \textbf{Reverse} setting within a custom monitor.

\textbf{Important}

\textit{You cannot use a monitor that includes the Transparent feature if you are using the IPv6 addressing or the route domains features.}

- **Transparent setting**
  Sometimes it is necessary to ping the aliased destination through a transparent pool, pool member, or node. When you create a custom monitor and set the \textbf{Transparent} setting to \texttt{Yes}, the BIG-IP system forces the monitor to ping \textit{through} the pool, pool member, or node with which it is associated (usually a firewall) to the pool, pool member, or node. (That is, if there are two firewalls in a load balancing pool, the destination pool, pool member, or node is always pinged through the pool, pool member, or node specified; not through the pool, pool member, or node selected by the load balancing method.) In this way, the transparent pool, pool member, or node is tested: if there is no response, the transparent pool, pool member, or node is marked as \textbf{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 \texttt{10.10.10.53:80} through a transparent firewall \texttt{10.10.10.101:80}. To do this, you create a monitor called \texttt{http\_trans} in which you specify \texttt{10.10.10.53:80} as the monitor destination address, and set the \textbf{Transparent} setting to \texttt{Yes}. Then you associate the monitor \texttt{http\_trans} with the transparent pool, pool member, or node.

  This causes the monitor to check the address \texttt{10.10.53:80} through \texttt{10.10.101:80}. (In other words, the BIG-IP system routes the check of \texttt{10.10.53:80} through \texttt{10.10.101:80}.) If the correct response is not received from \texttt{10.10.53:80}, then \texttt{10.10.101:80} is marked \textbf{down}.

  For more information on associating monitors with pool members or nodes, see \textit{Associating monitors with pools and nodes}, on page 12-15.
◆ Reverse setting

With the Reverse setting set to Yes, the monitor marks the pool, pool member, or node 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 12.3 shows the monitors that contain the Transparent setting, the Reverse setting, or both.

<table>
<thead>
<tr>
<th>Monitor Type</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>Transparent</td>
</tr>
<tr>
<td>HTTP</td>
<td>Transparent</td>
</tr>
<tr>
<td>TCP Echo</td>
<td>Transparent</td>
</tr>
<tr>
<td>ICMP</td>
<td>Transparent</td>
</tr>
</tbody>
</table>

Table 12.3 Monitors that contain the Transparent or Reverse settings

Configuring the Manual Resume setting

By default, when a monitor detects that a resource (that is, a node or a pool member) is unavailable, the BIG-IP system marks the resource as down and routes traffic to the next appropriate resource as dictated by the active load balancing method. When the monitor next determines that the resource is available again, the BIG-IP system marks the resource as up and immediately considers the resource to be available for load balancing connection requests. While this process is appropriate for most resources, there are situations where you want to manually designate a resource as available, rather than allow the BIG-IP system to do that automatically. You can manually designate a resource as available by configuring the Manual Resume setting of the monitor.

For example, consider a monitor that you assigned to a resource to track the availability of an HTML file, index.html, for a web site. During the course of a business day, you decide that you need to restart the system that hosts the web site. The monitor detects the restart action and informs the BIG-IP system that the resource is now unavailable. When the system restarts, the monitor detects that the index.html file is available, and begins sending connection requests to the web site. However, the rest of the web site might not be ready to receive connection requests. Consequently, the BIG-IP system sends connection requests to the web site before the site can respond effectively.
Configuring Monitors

To prevent this problem, you can configure the **Manual Resume** setting of the monitor. When you set the **Manual Resume** setting to **Yes**, you ensure that the BIG-IP system considers the resource to be unavailable until you manually enable that resource.

To summarize, if you set the **Manual Resume** setting of a monitor to **Yes** and then associate the monitor with a resource, and the resource subsequently becomes unavailable, the resource remains offline until you manually re-enable it.

**Modifying the Manual Resume setting of an existing monitor**

You can configure the **Manual Resume** setting when you create a custom monitor or you can modify the **Manual Resume** setting of an existing custom monitor. For information on creating a custom monitor, see *Creating a custom monitor*, on page 12-8. For information on modifying the **Manual Resume** setting of an existing customer monitor, see the following procedure.

**To modify the Manual Resume property**

1. On the Main tab of the navigation pane, expand **Local Traffic** and then click **Monitors**. The main monitors screen opens.
2. Click the name of the appropriate monitor. The properties screen for the monitor opens.
3. From the **Configuration** list, select **Advanced**.
4. For the **Manual Resume** option, click **Yes**.
5. Click **Update**.

**Resuming connections**

If you have a resource (such as a pool member or node) that a monitor marked as **down**, and the resource has subsequently become available again, you must manually re-enable that resource if the monitor’s **Manual Resume** setting is set to **Yes**. Manually re-enabling the resource allows the BIG-IP system to resume sending connections to that resource.

The procedure for manually re-enabling a resource varies depending on whether the resource is a pool, a pool member, or a node. For more information, see Chapter 3, *Configuring Nodes*, and Chapter 4, *Configuring Load Balancing Pools*. 
Configuring the Check Until Up setting

When you implement both active and passive monitoring on a pool member, you can enable the Check Until Up setting on an active monitor. Enabling the Check Until Up setting when you have configured active and passive monitoring causes the active monitor to check the health of the pool member only until the pool member is determined to be up. Once the pool member is determined to be up, the BIG-IP system disables active health checks for the pool member, leaving it to the Inband monitor to discover a down pool member.

Table 12.4 shows the difference between enabling and disabling the Check Until Up feature on an active monitor when passive monitoring is configured.

<table>
<thead>
<tr>
<th>If Check Until Up is:</th>
<th>And the Inband monitor marks pool member as up:</th>
<th>And the Inband monitor marks pool member as down:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>The active monitor refrains from checking the health of the pool member until the Inband monitor marks the pool member as down.</td>
<td>The active monitor checks the health of the pool member until it reports the pool member as up. Then the Inband monitor is used to monitor the pool member.</td>
</tr>
<tr>
<td>Disabled</td>
<td>The active monitor checks the health of the pool member continually.</td>
<td>The active monitor checks the health of the pool member continually.</td>
</tr>
</tbody>
</table>

Table 12.4 Effect of Check Until Up setting on active monitor when passive monitoring is configured

**Important**

When using this setting in conjunction with an Inband monitor, you must set the Inband monitor Retry Time setting to 0.
Associating monitors with pools and nodes

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

You can associate a monitor with a server in any of these ways:

- **Monitor-to-pool association**
  This type of association associates 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 every member of the pool `my_pool`, thus ensuring that all members of that pool are checked.

- **Monitor-to-pool member association**
  This type of association associates a monitor with an individual pool member, that is, an IP address and service. In this case, the monitor checks only that pool member and not any other members of the pool. For example, you can create an instance of the monitor `http` for pool member `10.10.10.10:80` of `my_pool`.

- **Monitor-to-node association**
  This type of association associates a monitor with a specific node. In this case, the monitor checks only the node itself, and not any services running on that node. For example, you can create an instance of the monitor `icmp` for node `10.10.10.10`. In this case, the monitor checks the specific node only, and not any services running on that node. You can designate a monitor as the default monitor that you want the BIG-IP system to associate with one or more nodes. In this case, any node to which you have not specifically assigned a monitor inherits the default monitor.

Some monitor types are designed for association with nodes only, and not pools or pool members. Other monitor types are intended for association with pools and pool members only, and not nodes. Node-only monitors specify a destination address in the format of an IP address only, with no service port (for example, `10.10.10.2`). Conversely, monitors that you can associate with nodes, pools, and pool members specify a destination address in the format of an IP address and service port (for example, `10.10.10.2:80`). Therefore, when you use the Configuration utility to associate a monitor with a pool, pool member, or node, 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 or its members, since the `icmp` monitor is designed to check the status of a node itself and not any service running on that node.

When you associate a monitor with a server, the BIG-IP 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.

For more information on associating monitors with pools and pool members, see Chapter 4, *Configuring Load Balancing Pools*. For more information on associating monitors with nodes, see Chapter 3, *Configuring Nodes*. 
Managing monitors

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

**Note**

*You can manage only those monitors that you have permission to manage, based on your user role and partition access assignment.*

**To display a monitor**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Monitors**.
   The Monitors screen opens.
2. Click a monitor name.
   This displays the monitor settings and their values.

**To delete a monitor**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Monitors**.
   The Monitors screen opens.
2. Check the Select box for the monitor that you want to delete.
3. Click **Delete**.
   A confirmation message appears.
4. Click **Delete**.

**To enable or disable a monitor instance**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Monitors**.
   The Monitors screen opens.
2. Click a monitor name in the list.
3. On the menu bar, click **Instances**.
   This lists any existing monitor instances.
4. For the instance you want to manage, check the Select box.
5. Click **Enable** or **Disable**.
6. Click Update.
Note

Because instances of monitors are not partitioned objects, a user can enable or disable an instance of a monitor without having permission to manage the associated pool or pool member. For example, a user with the Manager role, who can access partition AppA only, can enable or disable monitor instances for a pool that resides in partition Common. However, that user cannot perform operations on the pool or pool members that are associated with the monitor. Although this is correct functionality, the user might not expect this behavior. You can prevent this unexpected behavior by ensuring that all pools and pool members associated with monitor instances reside in the same partition.
Configuring NATs

- Understanding network address translation
- Creating a NAT
- Managing NATs
Understanding network address translation

In some cases, you might want to allow a client on an external network to send a request directly to a specific internal node (thus bypassing the normal load balancing server selection). To send a request directly to an internal server, a client normally needs to know the internal node’s IP address, which is typically a private class IP address. Because private class IP addresses are non-routable, you can instead create a network translation address (NAT). A NAT provides a routable IP address that an external node can use to send traffic to, or receive traffic from, an internal node.

What is a NAT?

A NAT is an address translation object that instructs the BIG-IP system to translate one IP address in a packet header to another IP address. A NAT consists of a one-to-one mapping of a public IP address to an internal private class IP address.

You can use a NAT in two different ways:

- **To translate a private class destination address to a public address**
  When an external node sends traffic to the public IP address defined in a NAT, the BIG-IP system automatically translates that destination address to the associated private class IP address, which represents a specific node on the internal network. This translation is hidden from the external node that sent the traffic. For more information, see *Understanding NATs for inbound connections*, on page 13-2.

- **To translate a private class source address to a public address**
  You can also use a NAT to translate an internal node’s private class source IP address to a public IP address. This translation is hidden from the external node that receives the traffic. For more information, see *Understanding NATs for outbound connections*, on page 13-4.

To summarize, a NAT provides a routable address for sending packets to or from a node that has a private class IP address.

When you create a NAT, you can map only one private class IP address to a specific public IP address. That is, a NAT always represents a one-to-one mapping between a private class IP address and a public IP address. If you want to map more than one private class IP address (that is, multiple internal nodes) to a single public IP address, you can create a SNAT instead. For more information, see Chapter 14, *Configuring SNATs*.

To create a NAT, you can either use the Configuration utility or the bigpipe utility. For information on using the Configuration utility, see *Creating a NAT*, on page 13-6. For information on using the bigpipe utility, see the *Bigpipe Utility Reference Guide*. 

Note

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.

Tip

When you use a NAT to provide access to an internal node, all ports on that internal node are open. To mitigate this security risk, consider using a SNAT instead. For more information, see Chapter 14, Configuring SNATs.

The BIG-IP system can apply a NAT to either an inbound or an outbound connection.

Understanding NATs for inbound connections

With respect to NATs, an inbound connection is a connection that is initiated by a node on an external network, and comes into the BIG-IP system to a node on the internal network.

Without a NAT

Normally, traffic coming into the BIG-IP system is load balanced to a server in a pool, based on the load balancing method configured for that pool, in the following way:

- A client on an external network typically sends traffic to a virtual server on the BIG-IP system. The destination IP address in this case is the virtual server address.
- Upon receiving a packet, the virtual server typically translates that destination IP address to the IP address of a pool member, for the purpose of load balancing that packet.
- The pool member then sends its response back through the BIG-IP system, using a route specified in the server node’s routing table (ideally, a floating IP address assigned to an internal VLAN). On receiving the response, the BIG-IP system then performs the reverse translation; that is, the system translates the pool member’s actual source address to the virtual server address. This results in the source address in the response to the client being the virtual server address, which is the source address that the client expects to see.

This typical load balancing scenario ensures that for load balanced traffic, the client system never sees the internal private class IP address of an internal node.
With a NAT

If the client system wants to bypass the load balancing mechanism to send packets directly to a specific node on the internal network, the client needs a routable IP address to use to send packets to that server node.

A NAT solves this problem by providing a routable address that a client can use to make a request to an internal server directly. In this way, a NAT performs the same type of address translation that a virtual server performs when load balancing connections to pool members. In the case of a NAT, however, no load balancing occurs, because the client is sending a request to a specific node. The NAT translates the public destination IP address in the request to the private class IP address of the internal node.

When the server node sends the response, the BIG-IP system performs the reverse translation, in the same way that a virtual server behaves. (For more information on this reverse translation, see Without a NAT, on page 13-2.)

◆ Note

The BIG-IP system does not track NAT connections. Therefore, the public IP address that you define in a NAT cannot be the same address as a virtual address or SNAT address.

Viewing an example

Suppose a node on the internal network (such as a load balancing server) has a private class IP address of 172.16.20.3. You can create a NAT designed to translate a public destination address of your choice (such as 207.10.1.103) to the private class address 172.16.20.3. Consequently, whenever a node on the external network initiates a connection to the address 207.10.1.103, the BIG-IP system translates that public destination address to the private class address 172.16.20.3.

Figure 13.1 illustrates the address translation that occurs for an inbound connection.
In this example, the NAT provides a routable address for an external node to initiate a connection to an internal node.

When you create a NAT, you must define two settings: **NAT Address** and **Origin Address**. In our example:

- The NAT address is **207.10.1.103**, and the origin address is **172.16.20.3**.
- The connection is an inbound connection, meaning that the connection is being initiated from the external network, through the BIG-IP system, to the internal network.
- The BIG-IP system translates the NAT address to the origin address.
- The NAT address and the origin address are destination addresses.

### Understanding NATs for outbound connections

The previous section, *Understanding NATs for inbound connections*, on page 13-2, summarized how a BIG-IP system normally load balances incoming traffic, and translates the source IP address in a response back to the virtual address.

Sometimes, however, an internal node needs to initiate a connection, rather than simply respond to a request. When a node on an internal network initiates a connection, the connection is considered to be an **outbound** connection. In this case, because the outgoing packets do not represent a response to a load-balanced request, the packets do not pass through a virtual server, and therefore the system does not perform the usual source IP address translation.

Without a NAT, the source IP address is a non-routable address. With a NAT, however, the BIG-IP system translates the internal node’s private class IP address to a public IP address, to which the external node can then route its response.

### Viewing an example

Suppose an internal node (such as a mail server) has a private class IP address of **172.16.20.1**. You can create a NAT designed to translate the private class address **172.16.20.1** to a public source address of your choice (such as **207.10.1.101**). Consequently, whenever the internal node **172.16.20.1** initiates a connection destined for a node on the external network, the system translates that source address of **172.16.20.1** to its public address (**207.10.1.101**).

Figure 13.2, on page 13-5, illustrates the address translation that occurs for an outbound connection.
In this example, the NAT provides a way for an internal node to initiate a connection to a node on an external network, without showing a private class IP address as the source address.

As previously mentioned, a NAT has two settings; **NAT Address** and **Origin Address**. In this example:

- The NAT address is **207.10.1.101**, and the origin address is **172.16.20.1**.
- The connection is an outbound connection, meaning that the connection is being initiated from the internal network, through the BIG-IP system, to the external network.
- The BIG-IP system translates the origin address to the NAT address.
- The origin address and the NAT address are source addresses.

As explained previously, a NAT always represents a one-to-one mapping between a public address and a private class address. However, if you would like to map multiple internal nodes to a single public address, you can use a secure network translation address (SNAT) instead of a NAT. You can use SNATs for outbound connections only. For more information, see Chapter 14, *Configuring SNATs*.

---

*Figure 13.2 Sample NAT for an outbound connection*
Creating a NAT

You must create a separate NAT address for each internal 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 of the navigation pane, expand **Local Traffic**, and click **SNATs**. The SNATs screen opens.
2. On the menu bar, click **NAT List**.
3. In the upper right corner, click **Create**. The New NAT screen opens.
   
   **Note:** If the **Create** button is unavailable, this indicates that your user role does not grant you permission to create a NAT.
4. In the **NAT Address** box, type the IP address that you want to use as the public, routable address for an internal node.
5. In the **Origin Address** box, type the private class IP address of the internal node.
6. Retain or modify all other values as necessary.
7. Click **Finished**.

Table 13.1 shows the settings that you can configure for a NAT, with a description of each.

<table>
<thead>
<tr>
<th>NAT Attribute</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT Address</td>
<td>An external, routable IP address on the BIG-IP system that you want external nodes to send connections to (for inbound connections) or receive connections from (for outbound connections).</td>
<td>No default value</td>
</tr>
<tr>
<td>Origin Address</td>
<td>The private class IP address of the internal node to which you want nodes on an external network to have access.</td>
<td>No default value</td>
</tr>
<tr>
<td>State</td>
<td>The state of the NAT, that is, whether the NAT is enabled or disabled.</td>
<td>Enabled</td>
</tr>
<tr>
<td>ARP</td>
<td>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.</td>
<td>Enabled</td>
</tr>
<tr>
<td>VLAN Traffic</td>
<td>The specific VLAN corresponding to the NAT address or origin address. VLANs to which the NAT is not to be mapped can be explicitly disabled, as when there is more than one internal VLAN.</td>
<td>All VLANS</td>
</tr>
</tbody>
</table>

Table 13.1  NAT configuration settings
Additional restrictions

When using a NAT, you should be aware of the following restrictions:

- The IP address defined in the **Origin Address** box must correspond to an actual, specific server on the internal network.
- You cannot reconfigure a NAT. Instead, you must delete the NAT and then re-create it.
Managing NATs

Using the Configuration utility, you can manage existing NATs in many ways. For example, you might want to modify the way that a standard NAT maps an original IP address to a translation address.

The tasks that you can perform when managing NATs are:

- Viewing or modifying a NAT
- Deleting a NAT
- Enabling or disabling a NAT for a load balancing pool

*Note*

To manage NATs, you must have an appropriate user role assigned to your user account.

Viewing or modifying NATs

You can view or modify any NAT that you created previously.

**To view or modify a NAT**

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs. This displays a list of existing SNATs.
2. On the menu bar, click NAT List. This displays a list of existing NATs.
3. In the NAT column, click a NAT address. This displays the properties of the NAT.
4. Modify any settings and click Update.

Deleting a NAT

You can delete any existing NAT that you created previously.

**To delete a NAT**

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs. This displays a list of existing SNATs.
2. On the menu bar, click NAT List.
3. In the NAT column, locate the NAT you want to delete, and check the Select box to the left.
4. At the bottom of the screen, click **Delete**.  
   A confirmation box appears.

5. Click **Delete**.

Enabling or disabling NATs for a load balancing pool

When configuring a load balancing pool, you can specifically disable NAT translations on any connections that use that pool. By default, this setting is enabled. For more information, see Chapter 4, *Configuring Load Balancing Pools*. 
Configuring SNATs

- Understanding secure network address translation
- Creating a SNAT pool
- Creating a SNAT
- Managing SNATs
- SNAT examples
Understanding secure network address translation

A secure network address translation (SNAT) translates the source IP address within a connection to a BIG-IP system IP address that you define. The destination node then uses that new source address as its destination address when responding to the request.

For inbound connections, that is, connections initiated by a client node, SNATs ensure that server nodes always send responses back through the BIG-IP system, when the server’s default route would not normally do so. Because a SNAT causes the server to send the response back through the BIG-IP system, the client sees that the response came from the address to which the client sent the request, and consequently accepts the response.

For outbound connections, that is, connections initiated by a server node, SNATs ensure that the internal IP address of the server node remains hidden to an external host when the server initiates a connection to that host.

Important

We recommend that before implementing a SNAT, you read Chapter 13, Configuring NATs.

A SNAT is similar to a NAT, except for some key differences:

- A NAT allows you to map only one original address to a translation address, while a SNAT allows you to map multiple original addresses to a single translation address. You can even map all node addresses on your network to a single public IP address, in a single SNAT object.
- When using a NAT, all ports on the internal node are open. By contrast, a SNAT by default supports UDP and TCP only. This, too, makes a SNAT more secure than a NAT.
- The BIG-IP system tracks SNAT connections, which, in turn, allows SNATs and virtual servers to use the same public IP addresses. This is different from NAT connections, which the BIG-IP system does not track.
- You must explicitly enable a NAT on the internal VLAN where the internal node’s traffic arrives on the BIG-IP system. By default, a SNAT that you create is enabled on all VLANs.

To create a SNAT, you can use the Configuration utility, the Traffic Management shell (tmsh), or the bigpipe utility. Alternatively, you can write an iRule, depending on the type of SNAT you are creating.

For information on using tmsh, see the Traffic Management Shell (tmsh) Reference Guide. For information on using the bigpipe utility, see the Bigpipe Utility Reference Guide. For information on iRules™, see the F5 Networks DevCentral web site http://devcentral.f5.com, and Chapter 17, Writing iRules.
Understanding SNATs for client-initiated (inbound) connections

In the most common client-server network configuration, the BIG-IP system’s standard address translation mechanism ensures that server responses return to the client through the BIG-IP system, thereby reversing the original destination IP address translation. This typical network configuration is as follows:

- The server nodes are on the same subnet as the BIG-IP system.
- The client nodes are on a different subnet from the server nodes.
- The BIG-IP system is the default gateway for the server subnet.

However, there are atypical network configurations in which the standard BIG-IP system address translation sequence by itself does not ensure that server responses use the required return path. Examples of these atypical configurations are:

- **When clients and servers are on the same network**
  If you want to load balance requests to server nodes that are on the same network as the client nodes, you can create a SNAT so that server responses are sent back through the virtual server, rather than directly from the server node to the client node. Otherwise, problems can occur such as the client rejecting the response because the source of the response does not match the destination of the request. Known as virtual server bounceback, this SNAT configuration causes the source of the response to match the destination of the request, thus ensuring that the client node accepts the response. You can use this kind of configuration when you want to load balance requests from web servers to application servers on the same network.

- **When the default gateway of the server node is not the BIG-IP system**
  For various reasons, the server node’s default route cannot always be defined to be a route back through the BIG-IP system. Again, this can cause problems such as the client rejecting the response because the source of the response does not match the destination of the request. The solution is to create a SNAT. When the BIG-IP system then translates the client node’s source IP address in the request to the SNAT address, this causes the server node to use that SNAT address as its destination address when sending the response. This, in turn, forces the response to return to the client node through the BIG-IP system rather than through the server node’s default gateway.

- **When using the OneConnect feature**
  The BIG-IP system OneConnect feature allows client requests to re-use idle server-side connections. Without a SNAT, the source IP address in the server-side connection remains the address of the client node that initially established the connection, regardless of which other client nodes re-use the connection. Although this is not an issue for traffic routing, you might find it confusing when examining various types of system output. A SNAT solves this problem.
**Note**

Using a SNAT for inbound connections can impact the availability of ephemeral ports. This can lead to the SNAT being unable to process additional connections until some source ports become available.

Figure 14.1 shows a typical problem for client-initiated connections when the BIG-IP system is not defined as the server’s default gateway, and you have not configured a SNAT for inbound traffic.

![Diagram of network traffic flow](image)

**Figure 14.1** Client rejects response due to non-matching destination and source IP addresses

To prevent these problems, you can configure an inbound SNAT. An **inbound SNAT** translates the original client source IP address in a request to a BIG-IP system virtual server or BIG-IP system self IP address, forcing subsequent server response to return directly to the BIG-IP system. When an inbound SNAT is configured on the system, the BIG-IP system translates not only the destination IP address in the request (using the standard address translation mechanism), but also the source IP address in the request (using a SNAT).

Figure 14.2, on page 14-4, shows that by configuring a SNAT, you ensure that the response returns through the BIG-IP system instead of through the default gateway, thus ensuring that the client can accept the server response.
Understanding SNATs for server-initiated (outbound) connections

When an internal server initiates a connection to an external host, a SNAT can translate the private, source IP addresses of one or more servers within the outgoing connection to a single, publicly-routable address. The external destination host can then use this public address as a destination address when sending the response. In this way, the private class IP addresses of the internal nodes remain hidden from the external host.

More specifically, a SNAT for an outgoing connection works in the following way:

1. The BIG-IP system receives a packet from an original IP address (that is, an internal server with a private IP address) and checks to see if that source address is defined in a SNAT.

2. If the original 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 packet, with the SNAT translation address as the source address, to the destination host.
Figure 14.3 shows an example of an outgoing SNAT. In this example, the BIG-IP system causes three internal nodes, with the IP addresses 172.16.20.4, 172.16.20.5, and 172.16.20.6, to advertise the public IP address 207.10.1.102 as the source IP address in the three outgoing connections.

**Figure 14.3** Sample SNAT for multiple outgoing connections

**Understanding SNAT implementation**

When you create a SNAT, you map an original IP address to a translation address in one of several ways, depending on your needs:

- You can explicitly map one or more original IP addresses to a single translation address (shown in Figure 14.3, on page 14-5).
- You can use the SNAT automap feature.
- You can create a pool of translation addresses, known as a SNAT pool, and map one or more original IP addresses to that SNAT pool.
- You can create a SNAT pool and map all original IP addresses to that SNAT pool.

**Mapping original IP addresses to a single 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. You create this type of SNAT using the New SNAT screen of the Configuration utility. For information on implementing this type of SNAT, see *Creating a SNAT*, on page 14-9.

**Using the SNAT automap feature**

Another type of SNAT is one that uses a feature of the BIG-IP system called SNAT automap. The SNAT automap feature automatically selects one of the system’s self IP addresses (typically a floating self IP address of the egress VLAN), and maps it to the original IP address or addresses that you specify during SNAT creation. When you use this feature, you do not need to explicitly specify a translation address.
When automatically choosing a self IP address to map to the specified original IP address, the system gives preference to floating self IP addresses over static (non-floating) self IP addresses. This prevents any interruption in service when failover occurs. Note that if no floating self IP address is currently assigned to the egress VLAN, the system uses the floating IP address of a non-egress VLAN instead.

For information on implementing this type of SNAT, see *Creating a SNAT*, on page 14-9.

### Mapping original IP addresses to a SNAT pool

You can also create a SNAT by creating a pool of translation addresses, and then mapping one or more original IP addresses 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 *Creating a SNAT*, on page 14-9.

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.**
  You can map addresses to a SNAT pool using the New SNAT screen in the Configuration utility. For more information, see *Creating a standard SNAT*, on page 14-9.

- **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 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 14-12.

### 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 14-13.
Creating a SNAT pool

If you decide that you want 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 original IP addresses to a SNAT pool, on page 14-6.

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 Creating a SNAT, on page 14-9.

A SNAT pool has two settings that you must configure when you create it. Table 14.1 lists and describes these settings.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The unique name of the SNAT pool.</td>
<td>No default value</td>
</tr>
<tr>
<td>Member List</td>
<td>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.</td>
<td>No default value</td>
</tr>
</tbody>
</table>

*Table 14.1 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 14-11.

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 14-9.
- 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 14-12.
- 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 14-13.

**Note**

SNAT pools cannot reside in partitions. Therefore, a user’s ability to create and manage SNAT pools is defined by his user role, rather than by his partition-access assignment. For more information, see the TMOS™ Management Guide for BIG-IP® Systems.
To create a SNAT pool

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs.
   The SNATs screen opens.

2. On the menu bar, click SNAT Pool List.
   This displays a list of existing SNAT pools.

3. In the upper-right corner of the screen, click Create.
   
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a SNAT.

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.
Creating 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 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 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 Chapter 13, Configuring NATs.

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).

Note that if you want to use a SNAT pool for the translation addresses, you must create the SNAT pool prior to creating the SNAT itself. For more information, see Creating a SNAT pool, on page 14-7.
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.

◆ Note

**SNATs cannot reside in partitions. Therefore, a user’s ability to create and manage SNATs is defined by the user role, rather than by the user’s partition-access assignment. For more information, see the TMOSTM Management Guide for BIG-IP® Systems.**

**To create a standard SNAT**

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SNATs**.
   The SNATs screen opens.
2. In the upper-right corner of the screen, click **Create**.
   
   _Note:_ If the Create button is unavailable, this indicates that your user role does not grant you permission to create a SNAT.
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 14.2 shows the settings that you can configure for a SNAT. Following the table are detailed descriptions of each setting.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the unique name of the standard SNAT. Setting this property is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Translation</td>
<td>Depending on the value selected, specifies an individual IP address, a SNAT pool name, or the Automap option. Possible values are: <strong>IP Address</strong>, <strong>SNAT Pool</strong>, or <strong>Automap</strong>.</td>
<td>Automap</td>
</tr>
<tr>
<td>Origin</td>
<td>Specifies the original IP addresses to which you want to map a translation address or pool of translation or self IP addresses. Possible values are <strong>All Addresses</strong> or <strong>Address List</strong>.</td>
<td><strong>All Addresses</strong></td>
</tr>
<tr>
<td>VLAN Traffic</td>
<td>The VLAN to which you want the SNAT to apply. Possible values are: <strong>ALL VLANS</strong>, <strong>Enabled On</strong>, and <strong>Disabled On</strong>.</td>
<td><strong>ALL VLANS</strong></td>
</tr>
</tbody>
</table>

**Table 14.2 Properties of a standard SNAT**
Specifying a SNAT name

The most basic setting you can configure for a 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 IP addresses. For background information on translation addresses, see Understanding secure network address translation, on page 14-1.

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.

- **SNAT pool**
  Specifying this value allows you to specify an existing SNAT pool to which you want to map your original IP address. For information on SNAT pools and how to create them, see Creating a SNAT pool, on page 14-7. 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 14-18.

- **Automap**
  Similar to a SNAT pool, the SNAT automap feature allows you to map one or more original 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 self IP addresses as the translation addresses for the pool.

When you specify translation addresses or a SNAT pool, the BIG-IP system automatically assigns a set of properties to those translation addresses. You can use the default values for these properties, or you can change them to suit your needs. Table 14.3 lists and describes the properties of a translation address.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>The IP address that you want to designate as a translation address. This is a required setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>State</td>
<td>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.</td>
<td>Enabled</td>
</tr>
<tr>
<td>ARP</td>
<td>A setting that determines whether the BIG-IP system responds to ARP requests or sends gratuitous ARPs.</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

Table 14.3 Properties of a SNAT translation address
For more information on defining, viewing, or modify translation addresses, see Managing SNATs, on page 14-14.

Specifying original IP addresses

The Origin setting specifies the original 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 one or more original 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 14-8.
- 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

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Limit</td>
<td>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.</td>
<td>0</td>
</tr>
<tr>
<td>TCP Idle Timeout</td>
<td>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.</td>
<td>Indefinite</td>
</tr>
<tr>
<td>UDP Idle Timeout</td>
<td>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.</td>
<td>Indefinite</td>
</tr>
<tr>
<td>IP Idle Timeout</td>
<td>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.</td>
<td>Indefinite</td>
</tr>
</tbody>
</table>

Table 14.3 Properties of a SNAT translation address
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 17, 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 2, Configuring Virtual Servers.

**Note**

For an example of an intelligent SNAT, see Example 2 -- Establishing an intelligent SNAT, on page 14-19.

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.

For information on creating a SNAT pool, see Creating a SNAT pool, on page 14-7. For information on assigning a SNAT pool directly to a virtual server, see Chapter 2, Configuring Virtual Servers.
Managing SNATs

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 a SNAT pool
- Defining translation addresses for an existing SNAT
- Viewing or modifying translation addresses for an existing SNAT
- Deleting SNATs, SNAT pools, and translation addresses
- Enabling or disabling SNATs for a load balancing pool
- Enabling or disabling SNAT translation addresses

Viewing or modifying SNATs and SNAT pools

You can view or modify any SNATs, NATS, or SNAT pools that you created previously.

To view or modify a SNAT

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs.
   This displays a list of existing SNATs.
2. In the SNAT column, click a SNAT name.
   This displays the properties of the SNAT.
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 of the navigation pane, expand Local Traffic, and click SNATs.
   This displays a list of existing SNATs.
2. On the menu bar, click SNAT Pool List.
   This displays a list of existing SNAT pools.
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 you can view or modify any existing translation addresses the you defined previously.

To explicitly define translation addresses for an existing SNAT

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs.
2. On the menu bar, click SNAT Translation List. This displays any existing translation addresses.
3. In the upper-right corner of the screen, click Create.
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create a SNAT.
4. Retain or change all property settings.
5. Click Finished.

To view or modify translation addresses for an existing SNAT

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs. This displays a list of existing SNATs.
2. On the menu bar, click SNAT Translation List. This displays a list of existing translation addresses.
3. Click a translation address.
4. View or modify the displayed settings.
5. If you modified any settings, click Update.
Deleting SNATs, SNAT pools, and translation addresses

You can delete any existing SNAT, SNAT pool, or translation address that you created previously.

To delete a SNAT

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs.
   This displays a list of existing SNATs.
2. In the SNAT column, locate the SNAT you want to delete and check the Select box to the left.
3. At the bottom of the screen, click Delete.
   This displays a confirmation box.
4. Click Delete.

To delete a SNAT pool

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs.
   This displays a list of existing SNATs.
2. On the menu bar, click SNAT Pool List.
   This displays a list of existing SNAT pools.
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.
   This displays a confirmation box.
5. Click Delete.

To delete a translation address

1. On the Main tab of the navigation pane, expand Local Traffic, and click SNATs.
   This displays a list of existing SNATs.
2. On the menu bar, click SNAT Translation List.
   This displays a list of existing translation addresses.
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.
   This displays a confirmation box.
5. Click Delete.
Enabling or disabling SNATs or NATs for a load balancing pool

When configuring a load balancing pool, you can specifically disable SNAT translations on any connections that use that pool. By default, this setting is enabled. For more information, see Chapter 4, *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 of the navigation pane, expand Local Traffic, click SNATs.
2. On the menu bar, click SNAT Translation List.
3. Locate the translation address you want to enable or disable, and check the Select the box to the left.
4. At the bottom of the screen, click Enable or Disable.
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 Internet service provider (ISP) has a BIG-IP system and 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 14.4.

```plaintext
pool isp_pool {
  lb_method rr
  member 199.5.6.254:0
  member 207.8.9.254:0
}
```

*Figure 14.4 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 14.5, on page 14-19. Note that the BIG-IP system automatically designates the SNAT pool members as translation addresses.
Configuring SNATs

Finally, using the Configuration utility, the ISP creates a SNAT that maps each original IP address directly to the appropriate SNAT pool.

Example 2 -- Establishing an intelligent SNAT

If you want to base SNAT mapping on criteria other than the original 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 14.6.

```
pool out_pool {
  lb_method round_robin
  member 199.5.6.254:0
  member 207.8.9.254:0
}
```

*Figure 14.6 bigip.conf entries for a pool to be used in an intelligent SNAT*

Next, as shown in Figure 14.7, 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 14.7.

```
snatpool chat_snatpool {
  member 199.5.6.10
  member 199.5.6.11
  member 207.8.9.10
  member 207.8.9.11
}
```

*Figure 14.7 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 14.8.

```
snatpool smtp_snatpool {
  member 199.5.6.20
  member 207.8.9.20
}
```

*Figure 14.8 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 14.9.

```
snatpool other_snatpool { 
    member 199.5.6.30
    member 207.8.9.30
}
```

**Figure 14.9** 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 14.10, 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` 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 ) {
        snatpool chat_snatpool
    } else if ( TCP::local_port equals 25 ) {
        snatpool smtp_snatpool
    } else {
        snatpool other_snatpool
    }
    pool out_pool
}
```

**Figure 14.10** 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 14.11.

```
virtual 0.0.0.0:0 use rule my_iRule
```

**Figure 14.11** Assignment of an iRule to a wildcard virtual server
15

Configuring Traffic Classes

• Introducing traffic classes
• Creating a traffic class
• Configuring traffic class settings
• Managing traffic classes
Introducing traffic classes

The BIG-IP® local traffic management system includes a feature called traffic classes. A traffic class is a feature that you can use when implementing optimization profiles for modules such as the WAN Optimization Module.

A traffic class allows you to classify traffic according to a set of criteria that you define, such as source and destination IP addresses. In creating the traffic class, you define not only classification criteria, but also a classification ID. Once you have defined the traffic class and assigned the class to a virtual server, the BIG-IP system associates the classification ID to each traffic flow. In this way, the BIG-IP system can regulate the flow of traffic based on that classification.

When attempting to match traffic flows to a traffic class, the BIG-IP system uses the most specific match possible.

You configure a traffic class by using the BIG-IP Configuration utility to create the traffic class and assign the class to a virtual server.

◆ Note

An alternate way to configure traffic classes is to use either the bigpipe utility or tmsh. For more information, see the Bigpipe Utility Reference Guide or the Traffic Management Shell (tmsh) Reference Guide.

Creating a traffic class

Like most local traffic objects, traffic classes always reside in partitions. Therefore, before creating a traffic class, you should set the current administrative partition to the partition in which you want the traffic class to reside. This is either the partition containing the virtual server that will reference the traffic class, or partition Common.

For example, if the virtual server that is to reference the traffic class resides in partition A, then the traffic class you create can reside in either partition A or partition Common.

To create a traffic class

1. On the Main tab of the navigation pane, expand Local Traffic, and click Traffic Class.
   The Traffic Class screen opens.

2. In the upper-right corner of the screen, click Create.

3. Configure all settings as needed.
   For information on settings, see Configuring traffic class settings, on page 15-2, or see the online help.

4. Click Finished.
After you have created a traffic class, you must assign it to a virtual server. For information on virtual servers, see Chapter 2, *Configuring Virtual Servers*.

### Configuring traffic class settings

Table 15.1 describes the settings that you can configure for a traffic class.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the traffic class. Traffic class names are case-sensitive and may contain letters, numbers, and underscores (_) only. Reserved keywords are not allowed. This setting is required.</td>
<td>No default value</td>
</tr>
<tr>
<td>Classification</td>
<td>Specifies a text string that the system associates with data flows that match the traffic-class criteria.</td>
<td>No default value</td>
</tr>
<tr>
<td>Source Address</td>
<td>Specifies a source IP address that the system should use to identify traffic belonging to this class. Any traffic originating from this IP address could be subject to the defined rate class.</td>
<td>No default value</td>
</tr>
<tr>
<td>Source Mask</td>
<td>Specifies the network mask for the address specified in the Source Address setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Source Port</td>
<td>Specifies a port name or number that the system should use to identify traffic belonging to this class. Any traffic originating from this port could be subject to the defined rate class.</td>
<td>No default value</td>
</tr>
<tr>
<td>Destination Address</td>
<td>Specifies a destination IP address that the system should use to identify traffic belonging to this class. Any traffic destined for this IP address could be subject to the defined rate class.</td>
<td>No default value</td>
</tr>
<tr>
<td>Destination Mask</td>
<td>Specifies the network mask for the address specified in the Destination Address setting.</td>
<td>No default value</td>
</tr>
<tr>
<td>Destination Port</td>
<td>Specifies a port name or number that the system should use to identify traffic belonging to this class. Any traffic destined this port could be subject to the defined rate class.</td>
<td>No default value</td>
</tr>
<tr>
<td>IP Protocol</td>
<td>Specifies the IP protocol to be used as a criterion for classifying traffic. Possible values are TCP and UDP. For example, if you specify a value of TCP, traffic coming into the BIG-IP system over TCP could become a member of this class.</td>
<td>TCP</td>
</tr>
</tbody>
</table>

*Table 15.1  Settings for configuring a traffic class*
Managing traffic classes

Once you have created a traffic class, you can list existing traffic classes, view or modify the settings of a traffic class, or delete a traffic class.

To list existing traffic classes
1. On the Main tab of the navigation pane, expand Local Traffic, and click Traffic Class.
   This displays a list of existing traffic classes and their setting values.
2. View the list of traffic classes.

To view or modify a traffic class
1. On the Main tab of the navigation pane, expand Local Traffic, and click Traffic Class.
   This displays a list of existing traffic classes.
2. Click a traffic class name in the list.
   This displays the settings for that traffic class.
3. Retain or modify any setting values. For information on traffic class settings, see Configuring traffic class settings, on page 15-2.
4. Click Update.

To delete a traffic class
1. On the Main tab of the navigation pane, expand Local Traffic, and click Traffic Class.
   This displays a list of existing traffic classes.
2. Locate a traffic 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.
   This displays a screen to confirm the deletion.
4. Click Delete.
   This removes the traffic class.
Configuring Rate Shaping

- Introducing rate shaping
- Creating and implementing rate classes
- Configuring rate class settings
- Managing rate classes
Introducing rate shaping

The BIG-IP® local traffic management 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 Local Traffic 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.

◆ Note

Rate classes cannot reside in partitions. Therefore, a user’s ability to create and manage rate classes is defined by user role, rather than partition-access assignment. For more information, see the TMOSTM Management Guide for BIG-IP® Systems.

You can create a rate class using the BIG-IP Configuration utility.

◆ Note

An alternate way to configure rate classes is to use the bigpipe utility or tmsh. For more information, see the Bigpipe Utility Reference Guide or the Traffic Management Shell (tmsh) Reference Guide.

To create a rate class

1. On the Main tab of the navigation pane, expand Local Traffic, and click Rate Shaping.
   The Rate Classes screen opens.
2. In the upper-right corner of the screen, click Create.
   This displays the New Rate Class screen.
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 16-8.
   - 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 16-8.
4. Configure all settings as needed.  
   For information on settings, see Configuring rate class settings, on page 16-4, 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 2, 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 17, Writing iRules.
Chapter 16

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 16.1.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies a unique name for the rate class. Every rate class requires a name.</td>
<td>No default value</td>
</tr>
<tr>
<td>Base Rate</td>
<td>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.</td>
<td>No default value</td>
</tr>
<tr>
<td>Ceiling Rate</td>
<td>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 Specifying a burst size, on page 16-6.</td>
<td>Same as Base Rate</td>
</tr>
<tr>
<td>Burst Size</td>
<td>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 Specifying a burst size, on page 16-6.</td>
<td>0</td>
</tr>
<tr>
<td>Direction</td>
<td>Specifies the direction of traffic to which the rate class is applied. Any: Specifies that the system applies the rate class to both directions of traffic. Client: Specifies that the system applies the rate class to traffic going from server to client. Server: Specifies that the system applies the rate class to traffic going from client to server.</td>
<td>Any</td>
</tr>
<tr>
<td>Parent Class</td>
<td>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 Specifying a burst size, on page 16-6.</td>
<td>None</td>
</tr>
<tr>
<td>Shaping Policy</td>
<td>Specifies a shaping policy that includes customized values for drop policy and queue method.</td>
<td>None</td>
</tr>
<tr>
<td>Queue Method</td>
<td>Specifies the method that the rate class uses to queue and dequeue traffic. Allowed settings are pfifo and sfq.</td>
<td>Same as parent class if a parent class is specified; otherwise, sfq.</td>
</tr>
<tr>
<td>Drop Policy</td>
<td>Specifies when and how to drop packets, if required, when the traffic handling queue is full..</td>
<td>tail</td>
</tr>
</tbody>
</table>

Table 16.1 Settings for configuring a rate class
Before configuring the settings of a rate class, it is helpful to have a description of those settings.

**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 (_). 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. The sum of the base rates of all child rate classes attached to a parent rate class, plus the base rate of the parent rate class, cannot exceed the ceiling of the parent rate class. For this reason, F5 recommends that you always set the base rate of a parent rate class to 0 (the default value).

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.

**Note**

*These numbers are powers of 10, not powers of 2.*

**Specifying a ceiling**

The **Ceiling Rate** setting specifies the absolute limit at which traffic is allowed to flow when bursting or borrowing. You can specify the ceiling 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 the rate class is a parent rate class, the value of the ceiling defines the maximum rate allowed for the sum of the base rates of all child rate classes attached to the parent rate class, plus the base rate of the parent rate class.

**Note**

*A child rate class can borrow from the ceiling of its parent rate class. For more information, see **Specifying a parent class**, on page 16-8.*
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.*

### 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 child rate class to borrow unused bandwidth from the ceiling of the parent class. A child class can borrow unused bandwidth from the ceiling of 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 ceiling of 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 ceiling of 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 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 shaping policy

Specifies a shaping policy that includes customized values for drop policy and queue method. The default value is None.

You can create additional shaping policies using the bigpipe utility or the Traffic Management shell (tmsh).

### 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 methods:

- **Stochastic Fair Queue**
  
  *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.
◆ Priority FIFO
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.

Specifying a drop policy

A drop policy specifies when and how to drop packets, if required, when the traffic handling queue is full. The default value is tail.

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.

You can create additional drop policies using the bigpipe utility or the Traffic Management shell (tmsh).
Configuring Rate Shaping

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 of the navigation pane, expand Local Traffic, and click Rate Shaping.
   This displays a list of existing rate classes and their setting values.
2. View the list of rate classes.

**To view or modify a rate class**

1. On the Main tab of the navigation pane, expand Local Traffic, and click Rate Shaping.
   This displays a list of existing rate classes.
2. Click a rate class name in the list.
   This displays the settings for that rate class.
3. Retain or modify any setting values. For information rate class settings, see Configuring rate class settings, on page 16-4.
4. Click Update.

**To delete a rate class**

1. On the Main tab of the navigation pane, expand Local Traffic, and click Rate Shaping.
   This displays a list of existing rate classes.
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.
   This displays a screen to confirm the deletion.
4. Click Delete.
   This removes the rate class.
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

An iRule is a powerful and flexible feature within the BIG-IP® local traffic management system that you can use to manage your network traffic. 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 the F5 Networks DevCentral web site, http://devcentral.f5.com. Note that iRules must conform to standard Tcl grammar rules; therefore, for more information on 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 17.1 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 17.1 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.

Using a feature called the Universal Inspection Engine, you can write an iRule that searches either a header of a packet, or actual packet content, and then directs the packet based on the result of that search. iRules can also direct packets based on the result of a client authentication attempt.
iRules can direct traffic not only to specific pools, but also to individual pool members, including port numbers and URI paths, either to implement persistence or to meet specific load balancing requirements.

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. Examples of event declarations that can trigger an iRule are HTTP_REQUEST, which triggers an iRule whenever the system receives an HTTP request, and CLIENT_ACCEPTED, which triggers an iRule when a client has established a connection.

Figure 17.2 shows an example of an event declaration within an iRule.

```
when HTTP_REQUEST {
    if { ![HTTP::uri] contains "aol" } {
        pool aol_pool
    } else {
        pool all_pool
    }
}
```

Figure 17.2 Example of an event declaration within an iRule

For more information on iRule events, see Specifying events, on page 17-8.
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 17.1.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational operators</td>
<td>contains, matches, equals, starts_with, ends_with, matches_regex</td>
</tr>
<tr>
<td>Logical operators</td>
<td>not, and, or</td>
</tr>
</tbody>
</table>

Table 17.1 iRule operators

For example, you can use the contains operator to compare a variable operand to a constant. You do this by creating an if statement that represents the following: "If the HTTP URI contains aol, send to pool aol_pool." Figure 17.2, on page 17-2, shows an iRule that performs this action.

iRule commands

An iRule command within an iRule causes the BIG-IP system to take some action, such as querying for data, manipulating data, or specifying a traffic destination. The types of commands that you can include within iRules are:

- **Statement commands**
  These 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 17-11.

- **Commands that query or manipulate data**
  Some commands search for header and content data, while others perform data manipulation such as inserting headers into HTTP requests. An example of a query command is IP::remote_addr, which searches for and returns the remote IP address of a connection. An example of a data manipulation command is HTTP::header remove <name>, which removes the last occurrence of the named header from a request or response.

- **Utility commands**
  These commands are functions that are useful for parsing and manipulating content. An example of a utility command is decode_uri <string>, which decodes the named string using HTTP URI encoding and returns the result. For more information on using utility commands, see Utility commands, on page 17-11.
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 17.3 shows an example of this command.

```
when HTTP_REQUEST {
    set uri [HTTP::uri]
    if { $uri ends_with ".gif" } {
        pool my_pool
    } elseif { $uri ends_with ".jpg" } {
        pool your_pool
    }
}
```

Figure 17.3 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 17.4 shows an example of this command.

```
when HTTP_REQUEST {
    if { [HTTP::uri] ends_with ".gif" } {
        node 10.1.2.200 80
    }
}
```

Figure 17.4 Example of the node command within an iRule
Selecting a pool of cache servers

You can create an iRule that selects a server from a pool of cache servers. Figure 17.5 shows an iRule that selects a server from a pool of cache servers.

```
when HTTP_REQUEST
    # This line specifies the expressions that determine whether the BIG-IP system sends requests to the cache pool:
    if { [HTTP::uri] ends_with "html" or [HTTP::uri] ends_with "gif" } {
        pool cache_pool
        set key [crc32 [concat [domain [HTTP::host] 2] [HTTP::uri]]]
        set cache_mbr [persist lookup hash $key node]
        if { $cache_mbr ne "" } {
            # This line verifies that the request is not coming from the cache:
            if { [IP::addr [IP::remote_addr] equals $cache_mbr] }
            # This line sends the request from the cache to the origin pool:
            pool origin_pool
            return
        }
    }
    # These lines ensure that the persistence record is added for this host/URI:
    persist hash $key
    } else {
        pool origin_pool
    }
}
```

**Figure 17.5 Load balancing to a server in a cache pool**

Note that the BIG-IP system redirects URIs to a new cache member at the time that the BIG-IP system receives a request for the URI, rather than when the pool member becomes unavailable.

Redirecting HTTP requests

In addition to configuring an iRule to select a specific pool, you can also configure an iRule to redirect an HTTP request to a specific location, using the `HTTP::redirect` iRule command. The location can be either a host name or a URI.

For example, Figure 17.6 shows an iRule that is configured to redirect an HTTP response.

```
when HTTP_RESPONSE
    if { [HTTP::status] contains "404" } {
        HTTP::redirect "http://www.siterequest.com/
    }
}
```

**Figure 17.6 An iRule based on HTTP redirection**
Figure 17.7 shows an example of an iRule that redirects an HTTP request.

```
when HTTP_REQUEST {
  if { [HTTP::uri] contains "secure"} {
    HTTP::redirect "https://[HTTP::host][HTTP::uri]"
  }
}
```

*Figure 17.7 Another iRule based on HTTP redirection*

**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.

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.

For more information on implementing SNATs, see Chapter 14, *Configuring SNATs*. 
Creating iRules

You can create an iRule using the Configuration utility.

Important

When you create an iRule, the BIG-IP system places the iRule into your current administrative partition. For information on partitions, see the TMOSTM Management Guide for BIG-IP® Systems.

To create an iRule

1. On the Main tab of the navigation pane, expand Local Traffic, and click iRules.
   The iRules screen opens.
2. In the upper right corner, click Create.
   Note: If the Create button is unavailable, this indicates that your user role does not grant you permission to create an iRule.
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 2, Configuring Virtual Servers.

Understanding iRules and administrative partitions

You should be aware of certain iRule configuration concepts as they relate to administrative partitions:

- An iRule can reference any object, regardless of the partition in which the referenced object resides. For example, an iRule that resides in partition_a can contain a pool statement that specifies a pool residing in partition_b.
- You can remove iRule assignments only from virtual servers that reside in the current Write partition or in partition Common.
- Note that you can associate an iRule only with virtual servers that reside in the current Write partition or in partition Common.
- You can associate an existing iRule with multiple virtual servers. In this case, the iRule becomes the only iRule that is associated with each virtual server in the current Write partition. Because this command overwrites all previous iRule assignments, F5 does not recommend use of this command.
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 the appropriate profile.**
  For example, if your iRule includes the event declaration `HTTP_REQUEST`, then the BIG-IP system only evaluates the iRule if the virtual server references an `http` profile type.

**Note**

*When assigning an iRule that specifies the event HTTP_REQUEST, make sure that the virtual server references the appropriate profile type.*

For information on assigning iRules and profiles to virtual servers, see Chapter 2, *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`
- HTTP events, such as `HTTP_REQUEST`
- SSL events, such as `CLIENTSSL_HANDSHAKE`
- Authentication events, such as `AUTH_SUCCESS`

For a complete list of iRule events and their descriptions, see the F5 Networks DevCentral web site, [http://devcentral.f5.com](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 17.8 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`.

```plaintext
when CLIENT_ACCEPTED {
  if { [IP::addr [IP::remote_addr] equals 10.1.1.80] } {
    pool my_pool1
  }
}
```

*Figure 17.8 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 17.8 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 17.9, on page 17-10, 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.
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 the F5 Networks DevCentral web site, [http://devcentral.f5.com](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 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`. One of the commands in the `HTTP` namespace is `HTTP::header`.

For a complete list of namespaces for iRules commands, see the F5 Networks DevCentral web site, [http://devcentral.f5.com](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 recognize 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, 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.

For example, an HTTP profile might specify a certain buffer size to use for compressing HTTP data, but you might want to specify a different buffer size for a particular type of HTTP connection. In this case, you can include the command **HTTP::compress_buffer_size** in your iRule, specifying a different value than the value in the profile.
Enabling session persistence with iRules

Chapter 7, *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. For example, if you have assigned the msrdp profile to the virtual server, then the BIG-IP system applies those settings to every incoming Microsoft® Remote Desktop Protocol (RDP) connection.

There are cases, however, when you might want to enable persistence in a more granular way. For example, instead of using the ssl persistence profile, which acts on non-terminated SSL traffic only, you might want to persist sessions based on SSL certificate status that you insert into the header of an HTTP request. To do this, you write an iRule using the `HTTP::header` command and then assign the iRule to the virtual server. Whenever the BIG-IP system accepts an SSL request, the iRule inserts the certificate status as a header into the request, and persists the session based on that status.

The BIG-IP system includes a special iRule command, `persist`, for implementing the types of session persistence described in Chapter 7, *Enabling Session Persistence*. You simply type the `persist` command in your iRule, specifying a persistence type.

For example, you can write an iRule that enables persistence for SSL connections when a particular event occurs, basing the persistence on the session ID. Figure 17.10 shows an example of an iRule that you could write to do this:

```plaintext
when CLIENTSSL_HANDSHAKE {
    persist ssl
}
```

*Figure 17.10  Sample iRule for SSL persistence based on session ID*

You can use the `persist none, hash, srcaddr, destaddr, and uie` commands in any circumstance, even if a corresponding persistence profile is not configured and assigned to the virtual server. However, the `persist ssl, cookie, msrdp, and sip` commands require that you assign a corresponding persistence profile to the virtual server. Attempts to use these commands without a corresponding profile result in a run-time iRule error. For information on assigning a persistence profile to a virtual server, see Chapter 2, *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.

The BIG-IP system includes three pre-configured data groups: **private_net**, **images**, and **aol**.

To understand the usefulness of data groups, it is helpful to first understand the **matchclass** command and the **contains** operator.

---

**Note**

You can manage only those data groups that you have permission to manage, based on your user role and partition access assignment.

---

**WARNING**

Do not attempt to modify or delete any of the three pre-configured data groups (**private_net**, **images**, and **aol**). Doing so can produce adverse results.

---

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 17.11 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).

```plaintext
when CLIENT_ACCEPTED {
  if { ![matchclass [IP::remote_addr] equals $::aol] } {
    pool aol_pool
  } else {
    pool all_pool
  }
}
```

*Figure 17.11* 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 of the navigation pane, expand Local Traffic, and click iRules.
   The iRules 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 appears 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.

To create a string data group

1. On the Main tab of the navigation pane, expand Local Traffic, and click iRules.
   The iRules 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 appears 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 of the navigation pane, expand Local Traffic, click iRules.
   The iRules 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 appears 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 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**. 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).

**Important**

If you attempt to load a `bigip.conf` file that contains external data group meta-data, and the file was created prior to BIG-IP system version 9.4, the system generates an error. The meta-data for the external data group contains the keyword `extern`, which generates an error during the load process. On BIG-IP systems running version 9.4 or later, the `extern` keyword is no longer needed in the `bigip.conf` file.

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.

An external data group must reside in either the `/config` or the `/var/class` directory. The default location for storing external data groups is the `/config` directory.

To implement an external data group, you must first use a text editor to create a text file according to the format specified in Figure 17.12, on page 17-18. Then you can use the Configuration utility to create a data group of type (**External File**) that references the text file that you previously created.
To store data groups externally

1. On the Main tab of the navigation pane, expand Local Traffic, and click iRules.
   The iRules 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).

6. In the Path / Filename box, type a path and file name for the external location, for example, /var/class/my_address_group.
   This file name should match the name that you assigned to the data group itself.

   **Note:** If you are storing the data group in the /config directory (the default location), you do not need to specify /config in the path name. Simply typing the file name of the data group is sufficient.

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 17.12 shows the contents of the data group file /config/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 17.12** 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 a data group

1. On the Main tab of the navigation pane, expand Local Traffic, and click iRules.
   The iRules screen opens.

2. On the menu bar, click Data Group List.

3. Click the name of a data group.
   This displays the properties of that data group.
Managing data group members

Using the Configuration utility, you can add members to or delete members from an existing data group if that data group is stored in line.

If the data group is stored externally, you can manage its members using the bigpipe utility. For more information, see the Bigpipe Utility Reference Guide.

**To add members to a data group**

1. On the Main tab of the navigation pane, expand Local Traffic, and click iRules.
   The iRules screen opens.

2. On the menu bar, click Data Group List.

3. Click the name of a data group.
   This displays the properties of that data group.

4. In the records area, type an address, string, or integer in the appropriate box.

5. Click Add.

6. Click Update.
Health and Performance Monitors

- Monitor reference
Monitor reference

The following pages contain descriptions of the local traffic management health and performance monitors. These descriptions are presented by name, in alphabetical order.

For conceptual and procedure information about monitors in general, see Chapter 12, Configuring Monitors.

**Important**

When defining values for custom monitors, make sure you avoid using any values that are on the list of reserved keywords. For more information, see solution number 3653 (for 9.0+ systems) on the Ask F5℠ Knowledge Base, [https://support.f5.com](https://support.f5.com).
External

Using an External type of monitor, you can create your own monitor type. To do this, you create a custom External-type monitor and within it, specify a user-supplied monitor to run.

Figure A.1 shows the settings and default values of an External-type monitor.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Type</td>
<td>External</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>External Program</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Arguments</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Variables</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

*Figure A.1 An External-type custom monitor with default values*

The settings are described as follows:

◆ **Name**
   Specifies a unique name for the custom monitor, such as `my_external_monitor`.

◆ **Type**
   Specifies the type of monitor you are creating.

◆ **Interval**
   Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

◆ **Timeout**
   Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

◆ **Manual Resume**
   Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Check Until Up**
   Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*. 
◆ **External Program**
   It is the **External Program** setting that you use to specify the executable name of your user-supplied monitor program. An External-type monitor searches the directory `/usr/bin/monitors` for that monitor name.

◆ **Arguments**
   The **Arguments** setting allows you to specify any command-line arguments that are required.

◆ **Variables**
   This setting specifies the variables that an External monitor requires, namely a Name/Value pair.

◆ **Alias Address and Alias Service Port**
   The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*. 
The purpose of this monitor is to check the health of FirePass systems.

Figure A.2 shows the settings and default values of a FirePass-type monitor.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Type</td>
<td>External</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Cipher List</td>
<td>HIGH:!ADH</td>
</tr>
<tr>
<td>Max Load Average</td>
<td>12.0</td>
</tr>
<tr>
<td>Concurrency Limit</td>
<td>95</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Alias Address</td>
<td>*All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>*All Ports</td>
</tr>
</tbody>
</table>

*Figure A.2 A FirePass-type custom monitor with default values*

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_firepass_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. Also, if the node responds with a RESET packet, the system flags the node as **down** immediately, without waiting for the timeout interval to expire. Note that the **Timeout** setting should be three times the **Interval** setting, plus 1 second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*. 
Cipher List
If you do not specify a cipher list, the monitor uses the default cipher list
DEFAULT:+SHA:+3DES:+kEDH.

Max Load Average
Specifies the number that the monitor uses to mark the FirePass system
up or down. The system compares this setting to a one-minute average of
the FirePass system load. When the FirePass system-load average falls
within the specified Max Load Average value, the monitor marks the
FirePass system up. When the average exceeds the setting, the monitor
marks the system down. The default value is 12.0.

Concurrency Limit
Specifies the maximum percentage of licensed connections currently in
use under which the monitor marks the FirePass system up. As an
example, a setting of 95 percent means that the monitor marks the
FirePass system up until 95 percent of licensed connections are in use.
When the number of in-use licensed connections exceeds 95 percent, the
monitor marks the FirePass system down. The default value is 95.

User Name and Password
If there is no password security, you must use blank strings ["""" for the
Username and Password settings.

Alias Address and Alias Service Port
The Alias Address setting specifies the destination IP address that the
monitor checks, with the default value * All Addresses. For more
information, see Chapter 12, Configuring Monitors.
FTP

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 that once the file has been successfully downloaded, the BIG-IP system does not save it.

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

Figure A.3 shows the settings and default values of an FTP-type monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>FTP</td>
</tr>
<tr>
<td>Interval</td>
<td>10</td>
</tr>
<tr>
<td>Timeout</td>
<td>31</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Path/Filename</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Mode</td>
<td>Passive</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

Figure A.3 An FTP-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_ftp_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined
to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.

- **Alias Address** and **Alias Service Port**
  The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*. 
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 member. (Remember that you can apply an ICMP monitor to a node only and not a pool member.) Figure A.4 shows the settings and their values for the pre-configured gateway_icmp monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Gateway ICMP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

Figure A.4 The gateway_icmp pre-configured monitor

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as **my_gw_icmp_monitor**.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.
◆ **Transparent**
  The **Transparent** mode is an option for this type of monitor. When you set this mode to **Yes**, the monitor pings the node with which the monitor is associated. For more information about **Transparent** mode, see Chapter 12, *Configuring Monitors*.

◆ **Alias Address** and **Alias Service Port**
  The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.
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. Figure A.5 shows the settings of the pre-configured monitor `http`.

<table>
<thead>
<tr>
<th>Name</th>
<th>''</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>HTTP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Send String</td>
<td>GET /</td>
</tr>
<tr>
<td>Receive String</td>
<td>''</td>
</tr>
<tr>
<td>User Name</td>
<td>''</td>
</tr>
<tr>
<td>Password</td>
<td>''</td>
</tr>
<tr>
<td>Reverse</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

*Figure A.5 The `http` pre-configured monitor*

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_http_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined
to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

- **Send String** and **Receive String**
  This 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. The check is successful when the content matches the **Receive String** value. Note that you can specify the value of a response header as the **Receive String** value. For example, the value of the **Receive String** attribute can be **404 Object Not Found**.

- **User Name** and **Password**
  If there is no password security, you must use blank strings ["""] for the **Username** and **Password** settings.

- **Transparent** and **Reverse**
  Both **Transparent** and **Reverse** modes are options. For more information about **Transparent** and **Reverse** modes, see Chapter 12, Configuring Monitors.
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.

The BIG-IP system provides two pre-configured HTTPS monitors, https and https_443. Figure A.6 shows the settings of the pre-configured monitor https, and Figure A.7 shows the settings of the pre-configured https_443.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>HTTPS</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Send String</td>
<td>GET /</td>
</tr>
<tr>
<td>Receive String</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Cipher List</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Enabled</td>
</tr>
<tr>
<td>Client Certificate</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Reverse</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

**Figure A.6** The https pre-configured monitor

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>HTTPS_443</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Send String</td>
<td>GET /</td>
</tr>
<tr>
<td>Receive String</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Cipher List</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Enabled</td>
</tr>
<tr>
<td>Client Certificate</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Reverse</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>HTTPS</td>
</tr>
</tbody>
</table>

**Figure A.7** The https_443 pre-configured monitor
The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_https_monitor` or `my_https_443_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.

- **Send String** and **Receive String**
  This 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. The check is successful when the content matches the **Receive String** value. Note that you can specify the value of a response header as the **Receive String** value. For example, the value of the **Receive String** attribute can be **404 Object Not Found**.

- **Cipher List**
  If you do not specify a cipher list, the monitor uses the default cipher list `DEFAULT:+SHA:+3DES:+kEDH`.

- **User Name** and **Password**
  If there is no password security, you must use blank strings `""` for the **Username** and **Password** settings.)

- **Compatibility**
  When you set the **Compatibility** setting to **Enabled**, this sets the SSL options to **ALL**.

- **Client Certificate**
  You use the **Client Certificate** setting to specify a certificate file that the monitor then presents to the server.
◆ **Transparent** and **Reverse**
Both **Transparent** and **Reverse** modes are options. For more information about **Transparent** and **Reverse** modes, see Chapter 12, *Configuring Monitors*.

◆ **Alias Address** and **Alias Service Port**
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*. 
ICMP

Using an ICMP type of monitor, you can use Internet Control Message Protocol (ICMP) to make a simple node check. The check is successful if the monitor receives a response to an ICMP_ECHO datagram. Figure A.8 shows the settings and their values for the pre-configured monitor `icmp`.

| Name   | ""          |
| Type   | ICMP        |
| Interval | 5          |
| Timeout | 16         |
| Manual Resume | No      |
| Check Until Up | No  |
| Transparent | No           |
| Alias Address | * All Addresses |

**Figure A.8** The `icmp` pre-configured monitor

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_icmp_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*. 
◆ **Transparent**  
The **Transparent** mode is an option for this type of monitor. When you set this mode to **Yes**, the monitor pings the node with which the monitor is associated. For more information about **Transparent** mode, see Chapter 12, *Configuring Monitors*.

◆ **Alias Address**  
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*. 
IMAP

With an IMAP type of monitor, you can check the status of Internet Message Access Protocol (IMAP) traffic. An IMAP monitor is essentially a POP3 type of monitor with the addition of the Folder setting. The check is successful if the monitor is able to log into a server and open the specified mail folder.

Figure A.9 shows the settings and default values of an IMAP-type monitor.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Type</td>
<td>IMAP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Folder</td>
<td>INBOX</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
<tr>
<td>Debug</td>
<td>No</td>
</tr>
</tbody>
</table>

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_imap_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.
◆ **User Name and Password**
   An IMAP monitor requires that you specify a user name and password.

◆ **Folder**
   The Folder setting specifies the mail folder that the monitor attempts to open when the monitor logs into a server. The check is successful if the monitor is able to log on to the server and open the specified mail folder.

◆ **Alias Address and Alias Service Port**
   The Alias Address setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Debug**
   The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes. The default setting is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes setting specifies that the system redirects error messages and additional information to the `/var/log/<monitor_type>_<ip_address>.<port>.log` file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.

◆ **Note**

   *Servers to be checked by an IMAP monitor typically require special configuration to maintain a high level of security, while also allowing for monitor authentication.*
Inband

With an Inband type of monitor, the BIG-IP system performs passive monitoring as part of client requests. Replacing the need to write an iRule to perform passive monitoring, an Inband monitor works with either a Standard or a Performance (Layer 4) type of virtual server.

When you configure an Inband type of monitor, the monitor, when a client, attempts to connect to a pool member and behaves as follows:

- If the pool member does not respond to a connection request after a user-specified number of tries within a user-specified time period, the monitor marks the pool member as down.
- After the monitor has marked the pool member as down, and after a user-specified amount of time has passed, the monitor tries again to connect to the pool member (if so configured).

A special strength of the F5 implementation is that an Inband type of monitor can be combined with active monitors. This type of monitor adds very little system overhead.

**Note**

For more information on passive monitoring, see Chapter 12, *Configuring Monitors*.

Figure A.10 shows the settings and default values for an Inband-type monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Inband</td>
</tr>
<tr>
<td>Failures</td>
<td>3</td>
</tr>
<tr>
<td>Failure Interval</td>
<td>30</td>
</tr>
<tr>
<td>Response Time</td>
<td>10</td>
</tr>
<tr>
<td>Retry Time</td>
<td>300</td>
</tr>
</tbody>
</table>

*Figure A.10* An Inband-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_inband_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Failures**
  The Failures setting specifies the number of failed responses that a pool member may send in the defined failure interval before the monitor marks the pool member down. The total number of failures can be any combination of failed connection attempts or failures to return data within the interval specified in the Response Time setting. The default value is 3. Note that systems with multiple tmm processes use a per-process number to calculate failures, depending on the specified load.
balancing method. For example, for the Round Robin load balancing method, if there are $N$ tmm processes and $M$ pool members, and the Failures setting is set to $L$, then up to $N \times M \times L + 1$ failures can occur before the system marks the node as down.

◆ Failure Interval
The Failure Interval setting specifies an interval, in seconds. If the given number of failures specified in the Failures setting occurs within this interval, the system marks the pool member as being unavailable.

◆ Response Time
The Response Time setting specifies an amount of time, in seconds. If the pool member does not respond with data after this amount of time has passed, the number of failures in this interval increments by 1. Specifying a value of 0 disables this setting.

◆ Retry Time
The Retry Time setting specifies the amount of time in seconds after the pool member has been marked unavailable before the system tries again to connect to the pool member. A successful connection causes the monitor to mark the pool member as up. Specifying a value of 0 disables this setting.

Virtual server considerations

Before implementing an Inband monitor, you should be aware of the following facts:

- An Inband monitor can be associated with these two types of virtual servers only: Standard and Performance (Layer 4).
- The virtual server Protocol setting must be set to TCP.
- The virtual server HTTP Profile setting must be set to either http, None, or the name of a custom HTTP profile.

Table A.1, on page A-21 shows the differences in specific system behaviors that result from implementing an Inband monitor with three different virtual server configurations.
If the clustered multi-processing (CMP) feature is enabled on the BIG-IP system, the number of failures that can occur within the specified Inband monitor failure interval before a pool member is marked as **down** is computed on each **tmm** instance.

For example, if the **Failures** and **Failure Interval** settings of the Inband monitor are set to **3** and **30** respectively (the default values), then the number of failures that must occur before a pool member is marked as **down** differs depending on whether CMP is enabled on the system:

---

### CMP considerations

<table>
<thead>
<tr>
<th>Virtual server type</th>
<th>Behavioral considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance (Layer 4)</td>
<td>If you configure an Inband monitor and associate it with a Performance (Layer 4) type of virtual server, a passive failure occurs when:</td>
</tr>
<tr>
<td></td>
<td>• A connection cannot be established.</td>
</tr>
<tr>
<td></td>
<td>• A tmm instance receives a RESET from the server.</td>
</tr>
<tr>
<td></td>
<td>• The client does not receive a response from the server before the value of the Inband monitor <strong>Response Time</strong> setting expires (if the value is non-zero).</td>
</tr>
<tr>
<td></td>
<td>After a pool member has been marked <strong>down</strong> by the Inband monitor for the number of seconds specified in the <strong>Retry Time</strong> setting (if the value is non-zero), the pool member becomes eligible to receive a request. A pool member is marked <strong>up</strong> when the client is able to successfully establish and complete a connection to the pool member.</td>
</tr>
<tr>
<td>Standard with no HTTP profile</td>
<td>If you configure an Inband monitor and associate it with a Standard type of virtual server that does not reference an HTTP type of profile, a passive failure occurs when a connection is established with the pool member, or the pool member is responding to requests (or both), and the maximum retransmit times specified in the TCP profile are exceeded.</td>
</tr>
<tr>
<td></td>
<td>After a pool member has been marked <strong>down</strong> by the Inband monitor for the number of seconds specified in the <strong>Retry Time</strong> setting (if the value is non-zero), the pool member becomes eligible to receive a request. If the client is able to successfully establish a connection with the pool member, the pool member is marked <strong>up</strong> when the connection is completed.</td>
</tr>
<tr>
<td>Standard with HTTP profile</td>
<td>If you configure an Inband monitor and associate it with a Standard type of virtual server that references an HTTP type of profile, a passive failure occurs when:</td>
</tr>
<tr>
<td></td>
<td>• A connection is established with the pool member, or the pool member is responding to requests (or both), and the maximum retransmit times specified in the TCP profile are exceeded.</td>
</tr>
<tr>
<td></td>
<td>• The client does not receive a response from the server before the value of the Inband monitor <strong>Response Time</strong> setting expires (if set to non-zero).</td>
</tr>
<tr>
<td></td>
<td>• The connection is terminated before the value of the <strong>Response Time</strong> setting expires.</td>
</tr>
<tr>
<td></td>
<td>After a pool member has been marked <strong>down</strong> by the Inband monitor for the number of seconds specified in the <strong>Retry Time</strong> setting (if the value is non-zero), the pool member becomes eligible to receive a request. After successfully responding to a client request, the pool member is marked <strong>up</strong>.</td>
</tr>
</tbody>
</table>

| Table A.1 | Differences in Inband monitor behavior depending on type of virtual server |

---

**Table A.1** Differences in Inband monitor behavior depending on type of virtual server
• Without CMP (that is, only one instance of tmm is running), then three passive failures must occur within 30 seconds for the monitor to report the pool member as **down**.

• With CMP (for example, two instances of tmm are running), then three passive failures must occur on only one of the tmm instances for the monitor to report the pool member as **down**. However, if client requests are distributed in round robin fashion between the two tmm instances, then six client request failures must occur for the monitor to report a pool member as **down**.
An LDAP type of monitor checks the status of Lightweight Directory Access Protocol (LDAP) servers. The LDAP protocol implements standard X.500 for email directory consolidation. A check is successful if entries are returned for the base and filter specified. An LDAP monitor requires a user name, a password, and base and filter strings. Figure A.11 shows the settings and default values of an LDAP-type monitor.

![Figure A.11](image)

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_ldap_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, **Configuring Monitors**.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined
to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

◆ User Name and Password
The User Name setting specifies a distinguished name, that is, an LDAP-format user name.

◆ Base
The Base setting specifies the starting place in the LDAP hierarchy from which to begin the query.

◆ Filter
The Filter setting specifies an LDAP-format key of the search item.

◆ Security
The Security setting specifies the security protocol to be used. Acceptable values are SSL, TLS, or None.

◆ Mandatory Attributes
The Mandatory Attributes setting affects the way that the system conducts the filter search. When the value is No, the system performs a one-level search for attributes, and if the search returns no attributes, the node is reported as up. When the value is Yes, the system performs a subtree search, and if the search returns no attributes, the node is not reported as up.

◆ Alias Address and Alias Service Port
The Alias Address setting specifies the destination IP address that the monitor checks, with the default value * All Addresses. For more information, see Chapter 12, Configuring Monitors.

◆ Debug
The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes. The default setting is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes setting specifies that the system redirects error messages and additional information to the

/var/log/<monitor_type>_<_ip_address>_<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.

For an LDAP monitor to work properly, the BIG-IP system must be able to perform a reverse DNS lookup on the address of the LDAP or LDAPS node. This reverse lookup allows the BIG-IP system to check the host name of the node's address when it verifies the SSL certificate. An external DNS server does not work with this type of monitor.

The reverse DNS lookup requirement applies to both LDAP and LDAPS nodes, even though LDAP does not require the use of an SSL certificate.

◆ Note

You do not need to insert an entry for each LDAP server into the /etc/hosts file.
Module Score

A Module Score type of monitor enables global and local traffic management systems to load balance in a proportional manner to local traffic management virtual servers associated with the Web Accelerator and Application Security Manager modules. When you configure a Module Score type of monitor, the local traffic management system uses SNMP to pull the `gtm_score` values from the downstream virtual servers and set the dynamic ratios on the associated upstream local traffic management pool members or nodes.

More specifically, the Module Score monitor retrieves the `gtm_score` values from the virtual server and the `gtm_vs_score` values associated with the virtual server. Then, if a pool name is not specified, this monitor sets the dynamic ratio on the node that is associated with the virtual server.

The BIG-IP system uses the lowest non-zero value of the `gtm_vs_score` values to set the dynamic ratio. If all `gtm_vs_score` values are zero, then the `gtm_score` value is used to set the dynamic ratios. If you specify a pool name in the monitor definition, then the dynamic ratio is set on the pool member. Figure A.12 shows the settings and default values for a Module Score monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>''</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Module Score</td>
</tr>
<tr>
<td>Interval</td>
<td>10</td>
</tr>
<tr>
<td>Timeout</td>
<td>30</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>SNMP Community</td>
<td>public</td>
</tr>
<tr>
<td>SNMP Version</td>
<td>v2c</td>
</tr>
<tr>
<td>SNMP IP Address</td>
<td>''</td>
</tr>
<tr>
<td>SNMP Port</td>
<td>161</td>
</tr>
<tr>
<td>Pool Name</td>
<td>''</td>
</tr>
</tbody>
</table>

**Figure A.12**  A Module Score-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_module_score_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 30 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**.
◆ **Manual Resume**
Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

◆ **SNMP IP Address**
In configuring a Module Score monitor, you must explicitly specify a value for the Name and the SNMP IP Address settings.

◆ **Pool Name**
You need only specify a pool name if you want the system to set dynamic ratios on a pool member instead of on the associated node for the pool member.

◆ **Note**
If you want to distribute traffic to a cluster of WebAccelerator or Application Security Module virtual servers, you must create a separate custom Module Score monitor for each back-end local traffic management system.
MSSQL

You use an MSSQL type of monitor to perform service checks on Microsoft® SQL Server-based services such as Microsoft® SQL Server versions 6.5 and 7.0.

The BIG-IP system requires installation of a JDBC driver before performing the actual login. For more information, see Appendix B, Additional Monitor Considerations.

If you receive a message that the connection was refused, verify that the IP address and port number or service are correct. If you are still having logon trouble, see Chapter 12, Configuring Monitors.

The remainder of this section on MSSQL monitors describes prerequisite tasks, the default monitor settings, and troubleshooting tips.

Prerequisite tasks for MSSQL

Before using an MSSQL-type monitor, you must download a set of JDBC Java™ Archive (JAR) files and install them on the BIG-IP system. For more information, see Appendix B, Additional Monitor Considerations.

MSSQL monitor settings and their default values

Figure A.13 shows the settings and default settings of an MSSQL-type monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>''</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>mssql</td>
</tr>
<tr>
<td>Interval</td>
<td>30</td>
</tr>
<tr>
<td>Timeout</td>
<td>91</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Send String</td>
<td>''</td>
</tr>
<tr>
<td>Receive String</td>
<td>''</td>
</tr>
<tr>
<td>User Name</td>
<td>''</td>
</tr>
<tr>
<td>Password</td>
<td>''</td>
</tr>
<tr>
<td>Database</td>
<td>''</td>
</tr>
<tr>
<td>Receive Row</td>
<td>''</td>
</tr>
<tr>
<td>Receive Column</td>
<td>''</td>
</tr>
<tr>
<td>Count</td>
<td>0</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
<tr>
<td>Debug</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure A.13  An MSSQL-type custom monitor with default values
The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_mssql_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 30 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 91 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.

- **Send String**
  The **Send String** setting is optional and specifies a SQL query statement that the BIG-IP system should send to the server. Examples are `SELECT * FROM sales` and `SELECT FirstName, LastName FROM Employees`. If you configure the **Send String** setting, you can also configure the **Receive String**, **Receive Row**, and **Receive Column** settings.

- **Receive String**
  The **Receive String** setting is an optional parameter that specifies the value expected to be returned for the row and column specified with the **Receive Row** and **Receive Column** settings. An example of a **Receive String** value is **ALAN SMITH**. You can only configure this setting when you configure the **Send String** setting.

- **Database**
  The **Database** setting specifies the name of the data source on the Microsoft® SQL-based server. Examples are `sales` and `hr`.

- **Receive Row**
  The **Receive Row** setting is optional, and is useful only if the **Receive String** setting is specified. This setting specifies the row in the returned
table that contains the **Receive String** value. You can only configure this setting when you configure the **Send String** and **Receive String** settings.

- **Receive Column**
The **Receive Column** setting is optional and is useful only if the **Receive String** setting is specified. This setting specifies the column in the returned table that contains the **Receive String** value. You can configure this setting only when you configure the **Send String** and **Receive String** settings.

- **Alias Address** and **Alias Service Port**
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

- **Debug**
The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default setting is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** setting specifies that the system redirects error messages and additional information to the `/var/log/<monitor_type>_<ip_address>.<port>.log` file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.

### Troubleshooting MSSQL logins

If an MSSQL monitor cannot log in to the server, and you have checked that the specified IP address and port number or service are correct, try the following:

- **Verify that you can log in using another tool.**
For example, the server program Microsoft Windows® NT® SQL Server version 6.5 includes a client program named ISQL/w. This client program performs simple logins to SQL servers. Use this program to test whether you can log in to the server using the ISQL/w program.

- **Add logon accounts using the Microsoft® SQL Enterprise Manager.**
On the Microsoft® SQL Server, you can run the SQL Enterprise Manager to add login accounts. When first entering the SQL Enterprise Manager, you may be prompted for the SQL server that you want to manage.

You can register servers by entering the machine name, user name, and password. If these names are correct, the server becomes registered and you are then able to click an icon for the server. When you expand the subtree for the server, there is an icon for login accounts.

Beneath this subtree, you can find the SQL logins. Here, you can change passwords or add new logons by right-clicking the **Logins** icon. Click
this icon to access the **Add login** option. After you open this option, type the user name and password for the new login, as well as which databases the login is allowed to access. You must grant the **test** account access to the database you specify in the EAV configuration.
NNTP

You use an NNTP type of monitor to check the status of Usenet News traffic. The check is successful if the monitor retrieves a newsgroup identification line from the server. An NNTP monitor requires a newsgroup name (for example, `alt.cars.mercedes`) and, if necessary, a user name and password.

Figure A.14 shows the settings and default values of an NNTP-type monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>NNTP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Newsgroup</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
<tr>
<td>Debug</td>
<td>No</td>
</tr>
</tbody>
</table>

**Figure A.14** An NNTP-type custom monitor with default values

The settings are described as follows:

- **Name**
  - Specifies a unique name for the custom monitor, such as `my_nntp_monitor`.

- **Type**
  - Specifies the type of monitor you are creating.

- **Interval**
  - Specifies the frequency at which the system issues the monitor check. The default value, in seconds, is 5.

- **Timeout**
  - Specifies the number of seconds in which the node must respond to the monitor request. The default value, in seconds, is 16. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  - Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  - Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.
◆ **Alias Address** and **Alias Service Port**
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Debug**
The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default value is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** value specifies that the system redirects error messages and additional information to the `/var/log/<monitor_type>_<ip_address>_<port>.log` file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
Oracle

With an Oracle type of monitor, you can check the status of an Oracle database server. The check is successful if the monitor is able to connect to the server, log in as the indicated user, and log out.

Figure A.15 shows the settings and default values of an Oracle-type monitor.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Type</td>
<td>Oracle</td>
</tr>
<tr>
<td>Interval</td>
<td>30</td>
</tr>
<tr>
<td>Timeout</td>
<td>91</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Send String</td>
<td>GET /</td>
</tr>
<tr>
<td>Receive String</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Database</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Receive Row</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Receive Column</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
<tr>
<td>Debug</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure A.15  An Oracle-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_oracle_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 30 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 91 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined
to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

◆ Send String
The **Send String** setting specifies a SQL statement that the BIG-IP system should send to the Oracle server. An example is `SELECT * FROM sales`.

◆ Receive String
The **Receive String** setting is an optional parameter that specifies the value expected to be returned for a specific row and column of the table that the **Send String** setting retrieved. An example of a **Receive String** value is **SMITH**.

◆ Database
In an Oracle type of monitor, the **Database** setting specifies the name of the data source on the Oracle server. Examples are **sales** and **hr**.

◆ Receive Row
The **Receive Row** setting is optional, and is useful only if the **Receive String** setting is specified. This setting specifies the row in the returned table that contains the **Receive String** value.

◆ Receive Column
The **Receive Column** setting is optional and is useful only if the **Receive String** setting is specified. This setting specifies the column in the returned table that contains the **Receive String** value.

◆ Alias Address and Alias Service Port
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value **All Addresses**. For more information, see Chapter 12, Configuring Monitors.

◆ Debug
The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default setting is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** setting specifies that the system redirects error messages and additional information to the /

/var/log/<monitor_type>_.<ip_address>..<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
A POP3 type of monitor checks the status of Post Office Protocol (POP) traffic. The check is successful if the monitor is able to connect to the server, log in as the indicated user, and log out. A POP3 monitor requires a user name and password.

Figure A.16 shows the settings and default values of a POP3-type monitor:

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>POP3</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>User Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
<tr>
<td>Debug</td>
<td>No</td>
</tr>
</tbody>
</table>

**Figure A.16** A POP3-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_pop3_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.
- **Alias Address and Alias Service Port**
  The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

- **Debug**
  The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default value is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** value specifies that the system redirects error messages and additional information to the
  
  `/var/log/<monitor_type>_<ip_address><port>.log`

  file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
RADIUS

Using a RADIUS type of monitor, you can check the status of Remote Access Dial-in User Service (RADIUS) servers. The check is successful if the server authenticates the requesting user. A RADIUS monitor requires a user name, a password, and a shared secret string for the code number.

◆ Note

Servers to be checked by a RADIUS monitor typically require special configuration to maintain a high level of security while also allowing for monitor authentication.

Figure A.17 shows the settings and default values of a RADIUS-type monitor.

![Figure A.17 A RADIUS-type custom monitor with default values](image)

The settings are described as follows:

◆ Name
  Specifies a unique name for the custom monitor, such as `my_radius_monitor`.

◆ Type
  Specifies the type of monitor you are creating.

◆ Interval
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

◆ Timeout
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

◆ Manual Resume
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.
- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.

- **Alias Address and Alias Service Port**
  The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

- **Debug**
  The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default setting is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** setting specifies that the system redirects error messages and additional information to the `/var/log/<monitor_type>_<ip_address>.<port>.log` file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
RPC

Through the RPC monitor, you can use the BIG-IP system to check the availability of specific programs that reside on a Remote Procedure Call (RPC) server. This monitor uses the `rpcinfo` command to query the RPC server and verify the availability of a given program.

The RPC monitor contains two unique settings: **program** and **version**. The program setting specifies the program or application which the monitor needs to verify is available. The version setting is an optional setting that specifies an exact version number of that program. If you do not specify a version number, the monitor uses the `rpcinfo` command to verify that at least one version of the program is available. In addition, you can use the **mode** option of the RPC monitor to verify the availability of an RPC server using either TCP or UDP.

The resource that this monitor checks is considered available if it responds with a *ready and waiting* statement to the monitor query.

Figure A.18 shows the settings and default values of an RPC-type monitor.

```
Name ""
Type RPC
Import Settings rpc
Interval 10
Timeout 31
Manual Resume No
Check Until Up No
Mode TCP
Program <name>
Version Number <number>
Alias Address * All Addresses
Alias Service Port * All Ports
Debug No
```

**Figure A.18** An RPC-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_rpc_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within the set time period, the node is considered to be *up*. If the node does not respond within the set time period, the node is considered to be *down*. The Timeout value should be three times the Interval value, plus one second.
◆ **Manual Resume**
Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

◆ **Check Until Up**
Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

◆ **Mode**
You can use the Mode option of the RPC monitor to verify the availability of an RPC server using either TCP or UDP.

◆ **Program**
The Program setting specifies the program or application which the monitor needs to verify is available.

◆ **Version Number**
The Version Number setting is an optional setting that specifies an exact version number of that program. If you do not specify a version number, the monitor uses the rpcinfo command to verify that at least one version of the program is available.

◆ **Alias Address and Alias Service Port**
The Alias Address setting specifies the destination IP address that the monitor checks, with the default value * All Addresses. For more information, see Chapter 12, Configuring Monitors.

◆ **Debug**
The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes. The default value is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes value specifies that the system redirects error messages and additional information to the /var/log/<monitor_type>_<ip_address>_<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
Real Server

A Real Server type of monitor checks the performance of a pool, pool member, or node that is running the RealSystem Server data collection agent. The monitor then dynamically load balances traffic accordingly. Performance monitors are generally used with dynamic ratio load balancing. For more information on performance monitors and dynamic ratio load balancing, see Chapter 4, Configuring Load Balancing Pools, and Appendix B, Additional Monitor Considerations.

Note

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

The BIG-IP system provides a pre-configured Real Server monitor named real_server. Figure A.19 shows the settings and default values of the real_server monitor.

Table: The real_server pre-configured monitor

- **Name**: ""
- **Type**: Real Server
- **Interval**: 5
- **Timeout**: 16
- **Manual Resume**: No
- **Method**: GET
- **Command**: GetServerStats
- **Metrics**: ServerBandwidth:1.5, CPUPercentUsage, MemoryUsage, TotalClientCount
- **Agent**: Mozilla/4.0 (compatible: MSIE 5.0; Windows NT)

Like all pre-configured monitors, the real_server monitor cannot be modified by users. However, if you want to modify the Metrics setting, you can create a custom Real Server monitor, to which you can add metrics and modify metric values.

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_real_server_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not
respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

---

**Note**

*When creating a custom Real Server monitor, you cannot modify the values of the Method, Command, and Agent settings.*

Table A.2 shows the complete set of server-specific metrics and metric setting default values that apply to the **GetServerStats** command.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Default Coefficient</th>
<th>Default Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerBandwidth (Kbps)</td>
<td>1.0</td>
<td>10,000</td>
</tr>
<tr>
<td>CPUPercentUsage</td>
<td>1.0</td>
<td>80</td>
</tr>
<tr>
<td>MemoryUsage (Kb)</td>
<td>1.0</td>
<td>100,000</td>
</tr>
<tr>
<td>TotalClientCount</td>
<td>1.0</td>
<td>1,000</td>
</tr>
<tr>
<td>RTSPClientCount</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td>HTTPClientCount</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td>PNAClientCount</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td>UDPTransportCount</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td>TCPTransportCount</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td>MulticastTransportCount</td>
<td>1.0</td>
<td>500</td>
</tr>
</tbody>
</table>

*Table A.2 Metrics for a Real Server monitor*

The **metric coefficient** is a factor determining how heavily the metric’s value counts in the overall ratio weight calculation. The **metric threshold** is the highest value allowed for the metric if the metric is to have any weight at all. To understand how to use these values, it is necessary to understand how the overall ratio weight is calculated. The overall ratio weight is the sum of relative weights calculated for each metric. The relative weights, in turn, are based on three factors:

- The value for the metric returned by the monitor
- The coefficient value
- The threshold value
Given these values, the relative weight is calculated as follows:

\[ w = \frac{(\text{threshold-value})}{\text{threshold}} \times \text{coefficient} \]

You can see that the higher the coefficient, the greater the relative weight calculated for the metric. Similarly, the higher the threshold, the greater the relative weight calculated for any metric value that is less than the threshold. (When the value reaches the threshold, the weight goes to zero.)

Note that the default coefficient and default threshold values shown in Table A.2 are metric defaults, not monitor defaults. The monitor defaults take precedence over the metric defaults, just as user-specified values in the custom real_server monitor take precedence over the monitor defaults. For example, the monitor shown specifies a coefficient value of 1.5 for ServerBandwidth and no value for the other metrics. This means that the monitor uses the monitor default of 1.5 for the ServerBandwidth coefficient and the metric default of 1 for the coefficients of all other metrics. However, if a custom monitor my_real_server were configured specifying 2.0 as the ServerBandwidth coefficient, this user-specified value would override the monitor default.

Metric coefficient and threshold are the only non-monitor defaults. If a metric not in the monitor is to be added to the custom monitor, it must be added to the list of metrics for the Metrics setting. The syntax for specifying non-default coefficient or threshold values is:

<metric>:<coefficient | <*>:<threshold>
If your network employs IBM® Group Workload Managers for some of your network resources, you can employ the SASP monitor to verify the availability of these resources. This monitor uses the Server/Application State Protocol (SASP) to communicate with the Group Workload Manager. The monitor queries the Group Workload Manager for information on the current weights of each managed resource. These weights determine which resource currently provides the best response time. When the monitor receives this information from the Group Workload Manager (GWM), it configures the dynamic ratio option for the resources, allowing the BIG-IP system to select the most appropriate resource to respond to a connection request.

◆ Note

When you assign a SASP monitor, the monitor initially marks the resources as down. This change in status occurs because the GWM might not yet have information pertaining to its resources. As soon as the monitor receives the results of its query, it changes the status as needed. In most configurations, the monitor receives these results within a few seconds.

Figure A.20 shows the settings and default values of an SASP-type monitor.

| Name | "" |
| Type | SASP |
| GWM Interval | Automatic |
| GWM Address | 10.10.5.23 |
| GWM Service Port | 3000 |
| GWM Protocol | TCP |

Figure A.20 An SASP-type custom monitor with default values

The settings are described as follows:

◆ Name
  Specifies a unique name for the custom monitor, such as my_sasp_monitor.

◆ Type
  Specifies the type of monitor you are creating.

◆ GWM Interval
  The GWM Interval option determines the frequency at which the monitor queries the GWM. You can either set this option to Automatic, which instructs the monitor to use the interval GWM recommends; or you can select Specify, which allows you to supply an interval. If you select Specify, you can assign an interval between 10 and 600 seconds.

◆ GWM Address
  The GWM Address option specifies the IP address of the Group Workload Manager.
◆ **GWM Service Port**
The **GWM Service Port** option specifies the port through which the SASP monitor communicates with the Group Workload Manager.

◆ **GWM Protocol.**
The **GWM Protocol** option allows you to specify which protocol the monitor uses: either TCP or UDP.
Scripted

You use the Scripted type of monitor to generate a simple script that reads a file that you create. The file contains `send` and `expect` strings to specify lines that you want to send or that you expect to receive. For example, Figure A.21 shows a sample file that you could create, which specifies a simple SMTP sequence. Note that the lines of the file are always read in the sequence specified.

```plaintext
expect 220
send "HELO bigip1.somecompany.net\r\n"
expect "250"
send "QUIT\r\n"
```

**Figure A.21** Sample SMTP sequence file for the Scripted monitor

Using a Scripted monitor, you can then generate a script that acts on the above file. When the Scripted monitor script reads this file, the script examines each line, and if the line has no quotation marks (" "), the line is sent or expected to be received as is. If the line is surrounded by quotation marks, the script strips off the quotation marks, and examines the line for escape characters, treating them accordingly.

Figure A.22 shows the settings and default values of a Scripted-type monitor.

```plaintext
Name ""
Type Scripted
Import Settings scripted
Interval 10
Timeout 31
Manual Resume No
Check Until Up No
Filename <filename>
Alias Address * All Addresses
Alias Service Port * All Ports
Debug No
```

**Figure A.22** A Scripted-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_scripted_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within
the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

- **Filename**
  The Filename setting specifies the name of a file that you create. The file contains send and expect strings to specify lines that you want to send or that you expect to receive.

- **Alias Address and Alias Service Port**
  The Alias Address setting specifies the destination IP address that the monitor checks, with the default value * All Addresses. For more information, see Chapter 12, Configuring Monitors.

- **Debug**
  The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes. The default setting is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes setting specifies that the system redirects error messages and additional information to the /var/log/<monitor_type>_<ip_address>.<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.

**Note**

When you create a file containing send and expect strings, store the file in the directory /config/eav.
Appendix A

SIP

You use a SIP type of monitor to check the status of SIP Call-ID services. By default, this monitor type issues a SIP OPTIONS request to a server device. However, you can use alternative protocols instead: TCP, TLS, and SIPS (that is, Secure SIP).

The request that the monitor issues to a device is designed to identify the options that the server device supports. If the proper request is returned, the device is considered to be up and responding to commands.

Figure A.23 shows the settings and default values of a SIP-type monitor.

![Figure A.23 A SIP-type custom monitor with default values](image)

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_sip_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.
◆ **Manual Resume**
Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Check Until Up**
Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Mode**
Possible values for the **Mode** setting are **TCP**, **UDP**, **TLS**, and **SIPS**.

◆ **Cipher List**
For **TLS** and **SIPS** modes only, specifies the list of ciphers for this monitor.

◆ **Compatibility**
For **TLS** and **SIPS** modes only, specifies, when enabled, that the SSL options setting (in OpenSSL) is set to **ALL**.

◆ **Client Certificate**
For **TLS** and **SIPS** modes only, specifies a client certificate that the monitor sends to the target SSL server.

◆ **Client Key**
For **TLS** and **SIPS** modes only, specifies a key for a client certificate that the monitor sends to the target SSL server.

◆ **Additional Accepted Status Codes**
Possible values for the **Additional Accepted Status Codes** setting are **Any**, **None**, and **Status Code List**. By selecting **Status Code List**, you can specify one or more accepted status codes, in addition to status code **200**, that the system treats differently. Multiple status codes should be separated by spaces. Specifying **Any** indicates that all status codes are to be used to mark a server as **up**. If this setting and the **Additional Rejected Status Codes** setting show the same value, the **Additional Accepted Status Codes** setting takes precedence.

◆ **Additional Rejected Status Codes**
Possible values for the **Additional Rejected Status Codes** setting are **Any**, **None**, and **Status Code List**. By selecting **Status Code List**, you can specify one or more rejected status codes that the system treats differently. Multiple status codes should be separated by spaces. Specifying **Any** indicates that all status codes are to be used to mark a server as **down**. If this setting and the **Additional Accepted Status Codes** setting show the same value, the **Additional Accepted Status Codes** setting takes precedence.

◆ **Header List**
Using the **Header List** setting, you can add, edit, or delete a header, but you cannot delete a default header the SIP monitor uses. Also, for certain headers, the only way to augment a default header, such as to add a parameter to it, is to override the entire header.
◆ SIP Request
The SIP Request setting specifies the request line of the SIP message. Make sure to specify a complete SIP request line, minus the trailing \r\n characters. The system uses the response code to determine whether the server is up or down. The monitor performs a simple, customized query to a SIP server. The monitor does not establish connections, perform hand-shaking, or process SIP traffic or requests. It only sends a request to a server and looks at the response code and (aside from matching the response to the request) ignores the rest of the response. As a result, this monitor does not support requests such as INVITE, since the monitor does not enter into a dialog.

◆ Alias Address and Alias Service Port
The Alias Address setting specifies the destination IP address that the monitor checks, with the default value * All Addresses. For more information, see Chapter 12, Configuring Monitors.

◆ Debug
The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. The default setting is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes setting specifies that the system redirects error messages and additional information to the /var/log/<monitor_type>_<ip_address>.<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
With the SMB monitor, you can use the BIG-IP system to verify the availability of an SMB/CIFS (Server Message Block/Common Internet File System) server. You can employ this monitor to either check the availability of the server as a whole, the availability of a specific service on the server, or the availability of a specific file used by a service.

If you only specify the server using the **SMB/CIFS Server** setting, leaving the **Service Name** and **Path/Filename** settings empty, the monitor attempts to retrieve a list of available services from the SMB/CIFS server. If the monitor acquires a list, it marks the server as **available**.

Figure A.24 shows the settings and default values of a SMB-type monitor.

![Table](https://example.com/table.png)

**Figure A.24** A SMB-type custom monitor with default values

The settings are described as follows:

- **Name**
  - Specifies a unique name for the custom monitor, such as `my_smb_monitor`.

- **Type**
  - Specifies the type of monitor you are creating.

- **Interval**
  - Specifies the frequency at which the system issues the monitor check.
  - The default is 10 seconds.

- **Timeout**
  - Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**.
  - The **Timeout** value should be three times the **Interval** value, plus one second.
Appendix A

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

- **Path/Filename**
  Specifies a specific file associated with a service. The monitor uses the relative path to the service itself when attempting to locate the file. You are not required to specify a value for this option; however, if you elect to use this option, you must also specify a service using the Service Name setting.

- **SMB/CIFS Server**
  Specifies the NetBIOS name of the SMB/CIFS server for which the monitor checks for availability. You must specify a server for this monitor to function.

- **Service Name**
  Specifies a specific service on the SMB/CIFS for which you want to verify availability. You are not required to specify a service name.

- **Alias Address and Alias Service Port**
  The Alias Address setting specifies the destination IP address that the monitor checks, with the default value * All Addresses. For more information, see Chapter 12, Configuring Monitors.

- **Debug**
  Specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes. The default setting is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes setting specifies that the system redirects error messages and additional information to the /var/log/<monitor_type>_<ip_address>,<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
SMTP

An SMTP type of monitor checks the status of Simple Mail Transport Protocol (SMTP) servers. This monitor type is an extremely basic monitor that checks only that the server is up and responding to commands. The check is successful if the mail server responds to the standard SMTP HELO and QUIT commands.

Figure A.25 shows the settings and default values of an SMTP-type monitor.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Type</td>
<td>SMTP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Domain</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
<tr>
<td>Debug</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure A.25 An SMTP-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_smtp_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.
◆ **Domain**
  An SMTP-type monitor requires a domain name.

◆ **Alias Address** and **Alias Service Port**
  The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Debug**
  The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default setting is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** setting specifies that the system redirects error messages and additional information to the 
  
  /var/log/<monitor_type>_<ip_address>.<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
SNMP DCA

With an SNMP DCA type of monitor, you can check the performance of a server running an SNMP agent such as UC Davis, for the purpose of load balancing traffic to that server. With this monitor you can define ratio weights for CPU, memory, and disk use.

Performance monitors are generally used with dynamic ratio load balancing. For more information on performance monitors and dynamic ratio load balancing, see Chapter 4, Configuring Load Balancing Pools, and Appendix B, Additional Monitor Considerations.

◆ Note

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

The BIG-IP system provides a pre-configured SNMP DCA monitor named snmp_dca. Figure A.26 shows the settings and values of the snmp_dca pre-configured monitor.

```
Name "" 
Type SNMP DCA 
Interval 10 
Timeout 30 
Community Public 
Version v1 
Agent Type UCD 
CPU Coefficient 1.5 
CPU Threshold 80 
Memory Coefficient 1.0 
Memory Threshold 70 
Disk Coefficient 2.0 
Disk Threshold 90 
Variables ""
```

Figure A.26 The snmp_dca pre-configured monitor

Pre-configured monitors are not user-modifiable. Thus, if you want to change the values for the SNMP DCA monitor settings, you must create an SNMP DCA-type custom monitor. Possible values for the Version setting are v1, v2c, and Other. Possible values for the Agent Type setting are UCD, Win2000, and Other.

When configuring an SNMP DCA custom monitor, you can use the default CPU, memory, and disk coefficient and threshold values specified in the monitors, or you can change the default values. Optionally, you can specify coefficient and threshold values for gathering other types of data. Note that if the monitor you are configuring is for a type of SNMP agent other than UC Davis, you must specify the agent type, such as Win2000.

To understand how to use the coefficient and threshold values, it is necessary to understand how the overall ratio weight is calculated. The overall ratio weight is the sum of relative weights calculated for each metric. The relative weights, in turn, are based on three factors:
• The value for the metric returned by the monitor
• The coefficient value
• The threshold value

Given these values, the relative weight is calculated as follows:

\[ w = \frac{(\text{threshold-value})}{\text{threshold}} \times \text{coefficient} \]

You can see that the higher the coefficient, the greater the relative weight calculated for the metric. Similarly, the higher the threshold, the greater the relative weight calculated for any metric value that is less than the threshold. (When the value reaches the threshold, the weight goes to zero.)
SNMP DCA Base

You use an SNMP DCA Base type of monitor to check the performance of servers that are running an SNMP agent, such as UC Davis. However, you should use this monitor only when you want the load balancing destination to be based solely on user data, and not CPU, memory or disk use.

Figure A.27 shows the settings and default values of an SNMP DCA Base type of monitor.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>snmp_dca_base</td>
</tr>
<tr>
<td>Interval</td>
<td>10</td>
</tr>
<tr>
<td>Timeout</td>
<td>30</td>
</tr>
<tr>
<td>Community</td>
<td>Public</td>
</tr>
<tr>
<td>Version</td>
<td>v1</td>
</tr>
<tr>
<td>Variables</td>
<td>&quot;&quot;</td>
</tr>
</tbody>
</table>

Figure A.27 An SNMP DCA-type custom monitor with default values

Performance monitors are generally used with dynamic ratio load balancing. For more information on performance monitors and dynamic ratio load balancing, see Chapter 4, Configuring Load Balancing Pools, and Appendix B, Additional Monitor Considerations.

◆ Note

Unlike health monitors, performance monitors do not report the status of pool, pool member, or node.
SOAP

A SOAP monitor tests a Web service based on the Simple Object Access protocol (SOAP). More specifically, the monitor submits a request to a SOAP-based Web service, and optionally, verifies a return value or fault. Figure A.28 shows the settings and default values of a SOAP-type monitor.

```
Name "
Type soap
Interval 5
Timeout 16
Manual Resume No
Check Until Up No
User Name "
Password "
Protocol HTTP
URL Path "
Namespace "
Method "
Parameter Name "
Parameter Type bool
Parameter Value "
Return Type bool
Return Value "
Expect Fault No
Alias Address * All Addresses
Alias Service Port * All Ports
Debug No
```

**Figure A.28** A SOAP-type custom monitor with default values

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_soap_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check.
  The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. Also, if the target responds with a RESET packet, the system flags the target as **down** immediately, without waiting for the timeout interval to expire. The **Timeout** value should be three times the **Interval** value, plus one second.
◆ **Manual Resume**
   Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Check Until Up**
   Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Protocol**
   Possible values for the **Protocol** setting are **HTTP** and **HTTPS**.

◆ **Parameter Type**
   Possible values for the **Parameter Type** setting are: **bool**, **int**, **long**, and **string**.

◆ **Return Type**
   Possible values for the **Return Type** setting are: **bool**, **int**, **short**, **long**, **float**, **double**, and **string**.

◆ **Expect Fault**
   Possible values for the **Expect Fault** setting are **No** and **Yes**.

◆ **Alias Address** and **Alias Service Port**
   The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*.

◆ **Debug**
   The **Debug** setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the **Debug** setting are **No** and **Yes**. The default setting is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** setting specifies that the system redirects error messages and additional information to the **/var/log/<monitor_type>_<ip_address>.<port>.log** file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
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. Figure A.29 shows the settings for the pre-configured monitor **tcp**.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>TCP</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Send String</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Receive String</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Reverse</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

*Figure A.29 The tcp pre-configured monitor*

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_tcp_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*. 
◆ **Send String** and **Receive String**
This 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. The check is successful when the content matches the **Receive String** value.

◆ **Transparent** and **Reverse**
Both **Transparent** and **Reverse** modes are options. For more information about **Transparent** and **Reverse** modes, see Chapter 12, *Configuring Monitors*.

◆ **Alias Address** and **Alias Service Port**
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*. 
TCP Echo

With a TCP Echo type of monitor, you can verify Transmission Control Protocol (TCP) connections. The check is successful if the BIG-IP system receives a response to a TCP Echo message.

To use a TCP Echo monitor type, you must ensure that TCP Echo is enabled on the nodes being monitored. Figure A.30 shows the settings for the pre-configured monitor tcp_echo.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>TCP Echo</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Address</td>
<td>* All Addresses</td>
</tr>
</tbody>
</table>

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_tcp_echo_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

- **Check Until Up**
  Enabling the **Check Until Up** feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, *Configuring Monitors*. 
◆ **Transparent**
   The Transparent mode is an option for this type of monitor. When you set this mode to Yes, the monitor pings the node with which the monitor is associated. For more information about Transparent mode, see Chapter 12, Configuring Monitors.

◆ **Alias Address and Alias Service Port**
   The Alias Address setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, Configuring Monitors.
TCP Half Open

A TCP Half Open type of monitor performs a quick check on the associated service by sending a TCP SYN packet to the service. As soon as the monitor receives the SYN-ACK packet from the service, the monitor considers the service to be in an up state, and sends a RESET to the service instead of completing the three-way handshake.

Figure A.31 shows the settings for the pre-configured monitor tcp_half_open.

<table>
<thead>
<tr>
<th>Name</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>TCP Half Open</td>
</tr>
<tr>
<td>Interval</td>
<td>5</td>
</tr>
<tr>
<td>Timeout</td>
<td>16</td>
</tr>
<tr>
<td>Manual Resume</td>
<td>No</td>
</tr>
<tr>
<td>Check Until Up</td>
<td>No</td>
</tr>
<tr>
<td>Transparent</td>
<td>No</td>
</tr>
<tr>
<td>Alias Addresses</td>
<td>* All Addresses</td>
</tr>
<tr>
<td>Alias Service Port</td>
<td>* All Ports</td>
</tr>
</tbody>
</table>

Figure A.31 The tcp_half_open pre-configured monitor

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as my_tcp_half_open_monitor.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be up. If the node does not respond within the set time period, the node is considered to be down. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.
- **Transparent**  
The **Transparent** mode is an option for this type of monitor. When you set this mode to **Yes**, the monitor pings the node with which the monitor is associated. For more information about **Transparent** mode, see Chapter 12, *Configuring Monitors*.

- **Alias Address and Alias Service Port**  
The **Alias Address** setting specifies the destination IP address that the monitor checks, with the default value *All Addresses*. For more information, see Chapter 12, *Configuring Monitors*. 
UDP

You use a UDP type of monitor when the system is sending User Datagram Protocol (UDP) packets. Designed to check the status of a UDP service, a UDP-type monitor sends one or more UDP packets to a target pool, pool member, or node.

Figure A.32 shows the settings and default values of a UDP-type monitor.

| Name  | ""             |
| Type  | UDP            |
| Interval | 5             |
| Timeout   | 16            |
| Manual Resume | No           |
| Check Until Up | No          |
| Send String    | default send string |
| Send Packets  | 2             |
| Timeout Packets | 2             |
| Alias Address | * All Addresses |
| Alias Service Port | * All Ports |
| Debug        | No            |

**Figure A.32** A UDP-type custom monitor with default values

The settings are described as follows:

- **Name**
  - Specifies a unique name for the custom monitor, such as `my_udp_monitor`.

- **Type**
  - Specifies the type of monitor you are creating.

- **Interval**
  - Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  - Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The Timeout value should be three times the Interval value, plus one second.

- **Manual Resume**
  - Using the Manual Resume setting, you can manually designate a resource as being available. For more information, see Chapter 12, Configuring Monitors.

- **Check Until Up**
  - Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.
Timeout Packets
The value in seconds of the Timeout Packets setting should be lower than the value of the Interval setting.

Debug
The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes. The default setting is No, which specifies that the system does not redirect error messages and additional information related to this monitor. The Yes setting specifies that the system redirects error messages and additional information to the
/var/log/<monitor_type>_ip_address.<port>.log file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.

When using a UDP-type monitor to monitor a pool, pool member, or node, you must also enable another monitor type, such as ICMP, to monitor the pool, pool member, or node. Until both a UDP-type monitor and another type of monitor to report the status of the UDP service as up, the UDP service receives no traffic. See Table A.3 for details.

<table>
<thead>
<tr>
<th>If a UDP monitor reports status as</th>
<th>And another monitor reports status as</th>
<th>Then the UDP service is</th>
</tr>
</thead>
<tbody>
<tr>
<td>up</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>up</td>
<td>down</td>
<td>down</td>
</tr>
<tr>
<td>down</td>
<td>up</td>
<td>down</td>
</tr>
<tr>
<td>down</td>
<td>down</td>
<td>down</td>
</tr>
</tbody>
</table>

Table A.3 Determining status of the UDP service
You use a WAP monitor to monitor Wireless Application Protocol (WAP) servers. The common usage for the WAP monitor is to specify the **Send String** and **Receive String** settings only. The WAP monitor functions by requesting a URL and finding the string in the **Receive String** setting somewhere in the data returned by the URL response. Figure A.33 shows the settings and default values of a WAP-type monitor.

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_wap_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 10 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 31 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **Manual Resume**
  Using the **Manual Resume** setting, you can manually designate a resource as being available. For more information, see Chapter 12, *Configuring Monitors*.

![Figure A.33 A WAP-type custom monitor with default values](image)
◆ Check Until Up
   Enabling the Check Until Up feature causes the monitor to check the health of the pool member as usual, until the pool member is determined to be up. When the pool member is determined to be up, the BIG-IP system disables health checks for the pool member. For more information, see Chapter 12, Configuring Monitors.

◆ Send String
   The Send String setting specifies a URL.

◆ Receive String
   The Receive String setting specifies the string that the monitor should find, somewhere in the data returned by the URL response.

◆ Secret
   The Secret setting is the RADIUS secret, a string known to both the client and the RADIUS server, and is used in computing the MD5 hash.

◆ Accounting Node
   The Accounting Node setting specifies the RADIUS node. If this is a null string and RADIUS accounting has been requested (accounting port is non-zero), then the WAP server node is assumed to also be the RADIUS node.

◆ Accounting Port
   The Accounting Port setting implements RADIUS accounting, which is optional. To implement RADIUS accounting, you must set the accounting port to a non-zero value. In this case, the monitor assumes that RADIUS accounting is needed, and an accounting request is sent to the specified accounting node and port to start accounting. This is done before the URL is requested. After the successful retrieval of the URL with the correct data, an accounting request is sent to stop accounting.

◆ Server ID
   The Server ID setting specifies the RADIUS NAS-ID of the requesting server (that is, the BIG-IP system). It is a string used as an alias for the FQDN.

◆ Call ID
   The Call ID setting is an identifier similar to a telephone number, that is, a string of numeric characters. For testing purposes, this value is usually a string of eleven characters.

◆ Session ID
   The Session ID setting is a RADIUS session ID, used to identify this session. This is an arbitrary numeric character string, often something like 01234567.

◆ Framed Address
   The Framed Address setting is a RADIUS framed IP address. The setting has no special use and is usually specified simply as 1.1.1.1.

◆ Debug
   The Debug setting specifies whether the monitor sends error messages and additional information to a log file created and labeled specifically for the monitor. Possible values for the Debug setting are No and Yes.
The default setting is **No**, which specifies that the system does not redirect error messages and additional information related to this monitor. The **Yes** setting specifies that the system redirects error messages and additional information to the `/var/log/<monitor_type>_<ip_address>.<port>.log` file. You can use the log information to help diagnose and troubleshoot unsuccessful health checks.
**WMI**

A WMI type of monitor checks the performance of a pool, pool member, or node that is running the Windows Management Infrastructure (WMI) data collection agent, and then dynamically load balances traffic accordingly.

You generally use performance monitors such as a WMI monitor with dynamic ratio load balancing. For more information on performance monitors and dynamic ratio load balancing, see Chapter 4, *Configuring Load Balancing Pools*, and Appendix B, *Additional Monitor Considerations*.

**Note**

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

Figure A.34 shows the settings and default values of a WMI-type monitor.

```
Name ""  
Type wmi  
Interval 5  
Timeout 16  
User Name ""  
Password ""  
Method POST  
URL /scripts/f5Isapi.dll  
Command GetCPUInfo, GetDiskInfo, GetOSInfo  
Metrics LoadPercentage, DiskUsage, PhysicalMemoryUsage:1.5, VirtualMemoryUsage:2.0  
Agent Mozilla/4.0 (compatible: MSIE 5.0; Windows NT)  
Post RespFormat=HTML
```

*Figure A.34  A WMI-type custom monitor with default values*

Note that when creating a custom WMI monitor, the only default values that you are required to change are the null values for name, user name, and password. Also note that you cannot change the value of the *Method* setting.

The settings are described as follows:

- **Name**
  Specifies a unique name for the custom monitor, such as `my_wmi_monitor`.

- **Type**
  Specifies the type of monitor you are creating.

- **Interval**
  Specifies the frequency at which the system issues the monitor check. The default is 5 seconds.

- **Timeout**
  Specifies the number of seconds in which the node must respond to the monitor request. The default is 16 seconds. If the node responds within the set time period, the node is considered to be **up**. If the node does not
respond within the set time period, the node is considered to be **down**. The **Timeout** value should be three times the **Interval** value, plus one second.

- **User Name** and **Password**
  If there is no password security, you must use blank strings "" for the **Username** and **Password** settings.)

- **Command** and **Metrics**
  See the following table.

Table A.4 shows the complete set of commands and metrics that you can specify with the **Command** and **Metrics** settings. Also shown are the default metric values.

<table>
<thead>
<tr>
<th>Command</th>
<th>Metric</th>
<th>Default Coefficient</th>
<th>Default Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCPUInfo</td>
<td>LoadPercentage (%)</td>
<td>1.0</td>
<td>80</td>
</tr>
<tr>
<td>GetOSInfo</td>
<td>PhysicalMemoryUsage (%)</td>
<td>1.0</td>
<td>80</td>
</tr>
<tr>
<td>GetDiskInfo</td>
<td>DiskUsage (%)</td>
<td>1.0</td>
<td>90</td>
</tr>
<tr>
<td>GetPerfCounters</td>
<td>TotalKBytesPerSec</td>
<td>1.0</td>
<td>10,000</td>
</tr>
<tr>
<td></td>
<td>ConnectionAttemptsPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>CurrentConnections</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>GETRequestsPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>PUTRequestsPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>POSTRequestsPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>AnonymousUsersPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>CurrentAnonymousUsers</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>NonAnonymousUsersPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>CurrentNonAnonymousUser</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>CGIRequestsPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>CurrentCGIRequests</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>ISAPIRequestsPerSec</td>
<td>1.0</td>
<td>500</td>
</tr>
</tbody>
</table>

*Table A.4 WMI-type monitor commands and metrics*
### Table A.4 WMI-type monitor commands and metrics

<table>
<thead>
<tr>
<th>Command</th>
<th>Metric</th>
<th>Default Coefficient</th>
<th>Default Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrentISAPIRequests</td>
<td></td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td>GetWinMediaInfo</td>
<td>AggregateReadRate</td>
<td>1.0</td>
<td>10,000 Kbps</td>
</tr>
<tr>
<td></td>
<td>AggregateSendRate</td>
<td>1.0</td>
<td>10,000 Kbps</td>
</tr>
<tr>
<td></td>
<td>ActiveLiveUnicastStreams</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>ActiveStreams</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>ActiveTCPStreams</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>ActiveUDPStreams</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>AllocatedBandwidth</td>
<td>1.0</td>
<td>10,000 Kbps</td>
</tr>
<tr>
<td></td>
<td>AuthenticationRequests</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>AuthenticationsDenied</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>AuthorizationRequests</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>AuthorizationsRefused</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>ConnectedClients</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>ConnectionRate</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>HTTPStreams</td>
<td>1.0</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>HTTPStreamsReadingHeader</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>HTTPStreamsStreamingBody</td>
<td>1.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>LateReads</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>PendingConnections</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>PluginErrors</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>PluginEvents</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>SchedulingRate</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>StreamErrors</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>StreamTerminations</td>
<td>1.0</td>
<td>100</td>
</tr>
</tbody>
</table>
Table A.4 WMI-type monitor commands and metrics

<table>
<thead>
<tr>
<th>Command</th>
<th>Metric</th>
<th>Default Coefficient</th>
<th>Default Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDPResendRequests</td>
<td>1.0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>UDPResendsSent</td>
<td>1.0</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
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, Microsoft® 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® local traffic management system provides a special monitor plug-in file and a health or performance monitor for each type of server. 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 B.1 shows the required monitor plug-in and the corresponding performance monitor types.

<table>
<thead>
<tr>
<th>Server Type</th>
<th>Monitor plug-in</th>
<th>Monitor Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealServer Windows server</td>
<td>F5RealMon.dll</td>
<td>Real Server</td>
</tr>
<tr>
<td>RealServer UNIX server</td>
<td>f5realmon.so</td>
<td>Real Server</td>
</tr>
<tr>
<td>Windows server with WMI</td>
<td>F5Isapi.dll or F5.IsHandler.dll</td>
<td>WMI</td>
</tr>
<tr>
<td>Windows 2000 Server server</td>
<td>SNMP agent</td>
<td>SNMP DCA and SNMP DCA Base</td>
</tr>
<tr>
<td>UNIX server</td>
<td>UC Davis SNMP agent</td>
<td>SNMP DCA and SNMP DCA Base</td>
</tr>
</tbody>
</table>

Table B.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 system
- 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`.

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 12, Configuring Monitors.
- To associate the performance monitor with the pool member, see Chapter 4, Configuring Load Balancing Pools.
- To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 4, Configuring Load Balancing Pools.

### Implementing a WMI monitor

For Windows running Windows Management Instrumentation (WMI), the BIG-IP system provides a Data Gathering Agent for the server, either F5Isapi.dll (for IIS versions 5.0 and 6.0), or F5.IsHandler.dll (for IIS versions 6.0 and 7.0). Configuring a Windows platform for Dynamic Ratio load balancing consists of four tasks:

- Installing the Data Gathering Agent on the IIS server. For more information, see the procedures in this section.
- Configuring a WMI monitor on the BIG-IP system.
  To configure a WMI monitor, see Chapter 12, Configuring Monitors.
- Associating the monitor with the server to gather the metrics.
  To associate the custom monitor with the pool member, see Chapter 4, Configuring Load Balancing Pools.
- Creating or modifying the server pool to use the Dynamic Ratio load balancing method.
  To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 4, Configuring Load Balancing Pools.

The procedure for installing the Data Gathering Agent on an IIS server differs depending on whether the server is running IIS version 5.0, 6.0, or 7.0, and whether the Data Gathering Agent is the file F5Isapi.dll or the file F5.IsHandler.dll. Table B.2 shows each of the Data Gathering Agent files and the IIS versions that support each file.

<table>
<thead>
<tr>
<th>Data Gathering Agent</th>
<th>IIS version 5.0</th>
<th>IIS version 6.0</th>
<th>IIS version 7.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5Isapi.dll</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>F5.IsHandler.dll</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

*Table B.2  IIS version support for the Data Gathering Agent files*
Appendix B

The remainder of the section *Implementing a WMI monitor* contains the procedures for installing the two Data Gathering Agent files, `F5Isapi.dll` and `F5.IsHandler.dll`.

**Installing the Data Gathering Agent F5Isapi.dll**

You can install the file `F5Isapi.dll` on IIS versions 5.0 and 6.0.

**To install the Data Gathering Agent F5Isapi.dll 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. Clear Logvisits. (Logging of each visit to the agent quickly fills up the log files.)
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.dll 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 wish 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.
   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`.

Installing the Data Gathering Agent F5.IsHandler.dll

You can install the file `F5.IsHandler.dll` on IIS versions 6.0 and 7.0.

To install the Data Gathering Agent F5.IsHandler.dll on an IIS 6.0 server

1. Create a `scripts` directory under the directory C:\Inetpub. (C:\Inetpub\scripts).
2. Create a `bin` directory under the `scripts` directory (C:\Inetpub\scripts\bin).
3. Set the properties of the `scripts` directory to `scripts` and `executables`. 
4. Copy the file `F5.IsHandler.dll` to the directory `C:\Inetpub\scripts\bin`.

5. In the `C:\Inetpub\scripts` directory, create the file `web.config`. Figure B.1 shows an example of this file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.web>
    <httpHandlers>
      <clear />
      <add verb="*" path="F5Isapi.dll" type="F5.IsHandler" />
    </httpHandlers>
  </system.web>
</configuration>
```

*Figure B.1 Sample web.config file on an IIS server running version 6.0*

6. From the Start menu, choose Control Panel and double-click **Administration Tools**.

7. Double-click **Internet Information Services**. This opens the IIS Management Console.

8. Expand the name of the local computer.

9. Allow the file `ASP.NET v2.0<build_number>`:
   a) Select **Web Server Extensions**.
   b) Select `ASP.NET v2.0<build_number>`.
   c) Click **Allow**.

10. Create a new virtual directory named `scripts`:
    a) Expand **Websites** and `<Default Web Site>`.
    b) Right click `<Default Web Site>`, choose New, and choose Virtual Directory.
    c) Click **Next**.
    d) Type `scripts` for the alias and click **Next**.
    e) Type the directory you created in step 1 (`C:\Inetpub\scripts\`) and click **Next**.
    f) Click **Next** again.
    g) Click **Finished**.

11. Create an application pool for the file `F5.IsHandler.dll`:
    a) Right click **Application Pools**, choose New, and choose Application Pool.
b) Type **F5 Application Pool** in the **Application Pool ID** box and click **OK**.

You can now use the remaining steps to set up the Application Pool, Mappings, Directory Security, and ASP.NET to the **scripts** virtual directory.

12. Right click **scripts** and select **properties**.

13. Set up the application pool:
   a) Click the Virtual Directory tab.
   b) From the **Application Pool** list, select **F5 Application Pool**.

14. Set up the mappings:
   a) Click the **Configuration** button.
   b) On the Mappings tab of the Application Configuration screen, click **Add**.
   c) For the executable, type the file name
      
      ```
      C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll
      ```
   d) For the file name extension, type **.dll**.
   e) Clear the box for **Check that file exists** and click **OK**.
   f) On the Application Configuration screen, click **OK**.

15. Set up directory security:
   a) Click the Directory Security tab.
   b) Click the **Edit** button.
   c) Disable authentication by clicking the **Anonymous Access and Integrated Windows** box.
   d) Check the **Basic Authentication** box and click **OK**.

   **Note**: If you are not authenticating locally, you might need to set the default domain or realm.

16. Set up the ASP.NET program:
   a) Click the ASP.NET tab.
   b) From the ASP.NET version list, select **2.0.<buildnumber>** (for example **2.0.50727**).

17. On the **scripts** Properties page, click **OK**.

18. Set up access to the IIS metabase:
   a) Run the command `aspnet_regiis -ga <ASP.NETUsername>`.
   b) See the web site [http://support.microsoft.com/?kbid=267904](http://support.microsoft.com/?kbid=267904).
To install the Data Gathering Agent F5.IsHandler.dll on an IIS 7.0 server

1. Create a `scripts` directory under the directory `C:\Inetpub`. 
   `(C:\Inetpub\scripts)`.
2. Create a `\bin` directory under the `scripts` directory 
   `(C:\Inetpub\scripts\bin)`.
3. Copy the file `F5.IsHandler.dll` to the directory 
   `C:\Inetpub\scripts\bin`.
4. In the `C:\Inetpub\scripts` directory, create the file `web.config`. 
   Figure B.2 shows an example of this file.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
 <system.webServer>
  <handlers>
   <clear />
   <add name="F5IsHandler" path="F5Isapi.dll"
    verb="*" type="F5.IsHandler" modules="ManagedPipelineHandler"
    scriptProcessor="" resourceType="Unspecified"
    requireAccess="Script" preCondition="" />
  </handlers>
  <security>
   <authentication>
    <anonymousAuthentication enabled="false" />
   </authentication>
  </security>
 </system.webServer>
</configuration>
```

**Figure B.2** Sample `web.config` file on an IIS server running version 7.0

5. Allow anonymous authentication to be overridden by using the `appcmd` command to set the override mode in the machine-level `applicationHost.config` file. For example:

   `appcmd set config "Default Web Site/scripts" /section:anonymousAuthentication /overrideMode:Allow /commit:APPHOST`

6. Set up a new application pool for the file `F5.IsHandler.dll`:
   a) From the Start menu, choose Control Panel.
   b) Choose Administrative Tools
   c) Choose Internet Information Services (IIS) Manager.
   d) From **Connections**, expand `<MachineName>` (MachineName\UserName).
   e) Right click the Application Pools menu and choose Add Application Pool.
f) In the Name box, type F5 Application Pool.
g) Click OK.

7. Create a new application named scripts:
   a) Expand Web Sites and <MachineName>.
   b) Right click <MachineName> and choose Add Application.
   c) In the Alias box, type scripts.
   d) To change the application pool, click Select.
   e) For the physical path, type the directory you created in step 1 (C:\Inetpub\scripts).
   f) Click OK.

8. Change the Authentication setting to Basic Authentication:
   a) Select scripts.
   b) In the center pane, double click Authentication.
   c) Verify that the status of all items under Authentication is Disabled, except for the Basic Authentication item.
      To enable or disable an authentication item, right click the name and choose Enable or Disable.

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 12, Configuring Monitors.
- To associate the custom monitor with the pool member, see Chapter 4, Configuring Load Balancing Pools.
- To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 4, 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.

- **SNMP 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 12, Configuring Monitors.
- To associate a monitor with the pool, see Chapter 4, Configuring Load Balancing Pools.
- To set the load balancing method on the pool to the Dynamic Ratio method, see Chapter 4, 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. From an Internet browser, go to www.microsoft.com.
2. From the web site, access the Download Center.
3. Near the top of the page, from the Search list, verify that All Downloads is selected.
4. In the text field to the right of the Search list, type JDBC Driver.
5. Click Go. This displays a list of available drivers.
6. Depending on which service pack you are running, click one of the Microsoft SQL Server 2000 Driver for JDBC links.
   
   Note: The BIG-IP system does not support Microsoft SQL Server 2005.
7. Scroll down the page to the section titled Files in This Download.
8. In the file list, locate the file named mssqlserver.tar.
9. On the right side of the screen, click Download.
   
   Note: Verify that you are downloading the UNIX .tar file, and not a Windows package. If not, return to the previous screen.
10. When queried, click Save. This saves the file to disk.
11. Create a Linux directory, and move the .tar file to that directory.
12. Uncompress the file, using this command:
   
   tar xvf mssqlserver.tar
   
   Note: This command extracts some files, including: EULA.txt, install.ksh, msjdbc.tar, and read.me.
13. Install the extracted files as follows:
   
   a) At the command line prompt, type install.ksh.
   b) Follow the scripted instructions. This untars the msjdbc.tar file, creating several subdirectories.
   c) Locate the subdirectory lib. This directory contains three JAR files.
   d) Copy the three JAR files to the BIG-IP system directory /usr/bin/monitors/builtins/.
14. Recursively remove the Linux directory that you previously created. This step is optional.
After you install the JAR files, F5 recommends that you either reboot the BIG-IP system or run the following command:

```
/usr/bin/monitors/builtins/DB_monitor cmd quit
```

Rebooting the system causes the BIG-IP system to recognize the newly-installed JAR files the next time that you run an MSSQL monitor.
Working with the OpenSSL Utility

- Introducing OpenSSL for the BIG-IP system
- Creating a certificate for a client system
- Creating a certificate for a web site
- Working with certificate revocation
- Performing other certificate-related tasks
Introducing OpenSSL for the BIG-IP system

When you are using the BIG-IP system to terminate and initiate SSL traffic, you can use the OpenSSL utility to:

- Generate certificates for client systems.
- Generate certificates for web sites.
- Generate certificate revocation lists (CRLs).
- Perform other certificate-related tasks, such as viewing certificates and converting them to PEM format.

Creating a certificate for a client system

For client-side authentication between a client and a BIG-IP system, you can create a certificate for that client.

**To create a client certificate**

1. Access the BIG-IP system prompt.
2. Generate a client key. For example:
   ```bash
   openssl genrsa -rand .rand -out auser1.key 1024
   ```
3. Generate a client certificate request, using the previously-generated key. For example:
   ```bash
   openssl req -new -out auser1.req -key auser1.key
   ```
4. Generate a client certificate with or without the LDAP CRL distribution point. Note that you must use OpenSSL 0.9.8.x or a newer version to generate certificates with embedded distribution points that are dirname-based addresses. (dirname is a utility that strips off the trailing part of a file name, and the result is the path name of the directory that contains the file.)

   In the following example, the certificate is named `auser1.crt`.

   - To generate the client certificate with the LDAP CRL distribution point, use the `openssl x509` command, as in the following example:
     ```bash
     openssl x509 -req -in auser1.req -out auser1.crt \
     -CAkey bigmirror-ca.key -CA bigmirror-ca.crt \
     -days 300 -CAcreateserial -CAserial serial \
     -extensions crl_ext -extfile bigmirror-ca.ext
     ```
   - To generate the client certificate without the LDAP CRL distribution point, use the `openssl x509` command, as in the following example:
     ```bash
     openssl x509 -req -in auser1.req -out auser1.crt \
     -CAkey bigmirror-ca.key -CA bigmirror-ca.crt \
     -days 300 -CAcreateserial -CAserial serial
     ```
5. Create a PKCS12 file using the above key and certificate pairs. For example:

```
openssl pkcs12 -export -in auser1.crt -inkey auser1.key -out auser1.p12 -name "auser1 pkcs12"
```

Creating a certificate for a web site

For server-side authentication between a web site and a BIG-IP system, you can create a certificate for that web site.

**To create a certificate for a web site**

1. Access the BIG-IP system prompt.
2. Create a key. For example:

   `openssl genrsa -rand .rand -out www.test.net.key 1024`

3. Generate a certificate request using the key that you generated in step 1. For example:

   `openssl req -new -key www.test.net.key -out www.test.net.req`

4. Using the request that you generated in step 2, generate a certificate named for the web site.

   - If you want to generate the certificate with the LDAP CRL distribution point, use the `openssl x509` command, as in the following example:
     
     ```
     openssl x509 -req -in www.test.net.req -out www.test.net.crt -CAkey bigmirror-ca.key -CA bigmirror-ca.crt -days 300 -CAcreateserial -CAserial serial -extensions crl_ext -extfile bigmirror-ca.ext
     ```

   - If you want to generate the certificate without the LDAP CRL distribution point, use the `openssl x509` command, as in the following example:
     
     ```
     openssl x509 -req -in www.test.net.req -out www.test.net.crt -CAkey bigmirror-ca.key -CA bigmirror-ca.crt -days 300 -CAcreateserial -CAserial serial
     ```
Working with certificate revocation

You can use the OpenSSL utility to create a certificate revocation list (CRL). The BIG-IP system checks a CRL to see if a client or server certificate being presented for authentication has been revoked.

You can also use the utility to revoke a certificate.

To create a certificate revocation list

1. From the BIG-IP system prompt, create a configuration file for the serial or index option. For example:
   
   ```
   echo -e \\
   'default_ca=ca
\ndatabase=index.txt
\nserial=serial'
   > bigmirror-ca.config
   ```

2. From the BIG-IP system prompt, generate a CRL that expires in thirty days. For example:
   
   ```
   openssl ca -config bigmirror-ca.config -gencrl -crldays 30 -keyfile bigmirror-ca.key -cert bigmirror-ca.crt -out bigmirror-ca.crl
   ```

To revoke a certificate

Revoke a client certificate, using the openssl command from the BIG-IP system prompt. For example, to revoke the client certificate auser1.crt:

```
openssl ca -config bigmirror-ca.config -keyfile bigmirror-ca.key -cert bigmirror-ca.crt -revoke auser1.crt
```
Performing other certificate-related tasks

There are a number of other SSL-certificate-related tasks that you can perform, using the OpenSSL utility. You access this utility from the BIG-IP system prompt.

**To verify a certificate**

Use this command to verify a certificate:

```
openssl verify -CAfile bigmirror-ca.crt www.test.net.crt
```

**To view a CRL**

Use this command to view a CRL:

```
openssl crl -in bigmirror-ca.crl -text -noout
```

**To view certificate information**

Use this command to view certificate information:

```
openssl x509 -in www.test.net.crt -text -noout
```

**To convert a certificate to PEM format**

Use this command to convert a certificate from PKCS12 (.P12 or .PFX) format to PEM format:

```
openssl pkcs12 -in auser1.p12 -out auser1.pem
```

**To add a password to an RSA key**

Use this command to add a password to an RSA key:

```
openssl rsa -in auser1.key -out auser1-enc.key -des3 -passout pass:secret
```

**To strip a password from an RSA key**

Use this command to strip a password from an RSA key:

```
openssl rsa -in auser1-enc.key -out auser1.key -passin pass:secret
```
active monitoring

*Active monitoring* checks the status of a pool member or node on an ongoing basis, at a set interval. See also *passive monitoring* and *simple monitoring*.

active unit

In a redundant system, the active unit is the system that currently load balances connections. If the active unit in the redundant system fails, the standby unit assumes control and begins to load balance connections. See also *redundant system*.

application security class

An application security class is an HTTP Class profile that you configure from within Application Security Manager.

application template

An application template is a tool within the BIG-IP Configuration utility that eases the task of configuring the system to process application-specific traffic. An example of an application template is the template for Microsoft® SharePoint.

archive

An archive is a backup copy of the BIG-IP system configuration data. This archive is in the form of a user configuration set, or UCS. See also *user configuration set (UCS)*.

ARP (Address Resolution Protocol)

ARP is an industry-standard protocol that determines a host’s Media Access Control (MAC) address based on its IP address.

authentication

Authentication is the process of verifying a user’s identity when the user is attempting to log on to a system.

authentication iRule

An authentication iRule is a system-supplied or user-created iRule that is necessary for implementing a PAM authentication module on the BIG-IP system. See also *iRule, PAM (Pluggable Authentication Module)*.

authentication module

An authentication module is a PAM module that you create to perform authentication or authorization of client traffic. See also *PAM (Pluggable Authentication Module)*.
authentication profile
An authentication profile is a configuration tool that you use to implement a PAM authentication module. Types of authentication modules that you can implement with an authentication profile are: LDAP, RADIUS, TACACS+, SSL Client Certificate LDAP, and OCSP. See also PAM (Pluggable Authentication Module).

authorization
Authorization is the process of identifying the level of access that a logged-on user has been granted to system resources.

bigtop
The bigtop utility is a statistical monitoring utility that ships on the BIG-IP system. This utility provides real-time statistical information.

BIND (Berkeley Internet Name Domain)
BIND is the most common implementation of the Domain Name System (DNS). BIND provides a system for matching domain names to IP addresses. For more information, refer to http://www.isc.org/products/BIND.

BPDU (bridge protocol data unit)
A BPDU is a special packet that a spanning tree protocol sends between Layer 2 devices to determine redundant paths, and provide loop resolution. See also STP (Spanning Tree Protocol), RSTP (Rapid Spanning Tree Protocol), and MSTP (Multiple Spanning Tree Protocol).

bursting
Bursting is an aspect of rate shaping and occurs when the rate of traffic flow exceeds the base rate defined.

BYE transaction
A BYE transaction is a message that an application sends to another application when it is ready to close the connection between the two.

Call-ID
A Call-ID is a globally unique identifier that groups together a series of messages, which are sent between communicating applications.

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.

**certificate revocation list (CRL)**
See CRL (certificate revocation list).

**Certificate Revocation List Distribution Point (CRLDP)**
See CRLDP (Certificate Revocation List Distribution Point).

**certificate verification**
Certificate verification is the part of an SSL handshake that verifies that a client’s SSL credentials have been signed by a trusted certificate authority.

**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.

**chunking**
See HTTP chunking.

**cipher**
A cipher is an encryption/decryption algorithm that computer systems use when transmitting data using the SSL protocol.

**client-side SSL profile**
A client-side SSL profile is an SSL profile that controls the behavior of SSL traffic going from a client system to the BIG-IP system.

**clone pool**
This feature causes a pool to replicate all traffic coming into it and send that traffic to a duplicate pool.

**clustered multi-processing (CMP)**
See CMP (clustered multi-processing).

**CMP (clustered multi-processing)**
CMP is an internal performance-enhancement feature available on certain multi-processor BIG-IP systems. With CMP, the BIG-IP system creates multiple instances of the TMM service to process application traffic simultaneously.
configuration object
A configuration object is a user-created object that the BIG-IP system 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 synchronization
Configuration synchronization is the task of duplicating a BIG-IP system’s configuration data onto its peer unit in a redundant system.

Configuration utility
The Configuration utility is the browser-based application that you use to configure the BIG-IP system.

connection persistence
Connection persistence is an optimization technique whereby a network connection is intentionally kept open for the purpose of reducing handshaking. See also HTTP transformation and OneConnect™.

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 select a pool based on data contained within a packet.

cookie persistence
Cookie persistence is a mode of persistence where the BIG-IP system stores persistent connection information in a cookie.

CRL (certificate revocation list)
A CRL is a list that an authenticating system checks to see if the SSL certificate that the requesting system presents for authentication has been revoked.

CRLDP (Certificate Revocation List Distribution Point)
CRLDP is an industry-standard protocol that manages SSL certificate revocation for devices on a network.
CRLDP authentication module

A CRLDP authentication module is a user-created module that you implement on a BIG-IP system to authenticate client traffic using the CRLDP protocol. The purpose of a CRLDP authentication module is to manage the revocation of client SSL certificates on a network.

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 and profile.

data group

A data group is 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.

default profile

A default profile is a profile that the BIG-IP system 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 route

A default route is the route that the system uses when no other route specified in the routing table matches the destination address or network of the packet to be routed.

default VLAN

The BIG-IP system is configured with two default VLANs, one for each interface. One default VLAN is named internal and one is named external. See also VLAN (Virtual Local Area Network).

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.

**dynamic route**

A dynamic route is a route that an advanced routing protocol such as RIP adds dynamically to a routing table. See also static route.

**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 BIG-IP system. 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 BIG-IP system. See also health check.

**external authentication**

External authentication refers to the process of using a remote server to store data for the purpose of authenticating users or applications attempting to access the BIG-IP system.

**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.
**failback**

Failback is the process whereby an active unit relinquishes processing to a previously-failed unit that is now available.

**failover**

Failover is the process whereby a standby unit in a redundant system takes over when a software failure or a hardware failure is detected on the active unit.

**failover cable**

The failover cable directly connects the two units together in a redundant system.

**Fastest method**

Fastest mode is a load balancing method that passes a new connection based on the fastest response of all currently active nodes.

**FDDI (Fiber Distributed Data Interface)**

FDDI is a multi-mode protocol used for transmitting data on optical-fiber cables at speeds up to 100 Mbps.

**floating self IP address**

A floating self IP address is an additional self IP address for a VLAN that serves as a shared address by both units of a BIG-IP redundant system.

**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.

**gateway pool**

A gateway pool is a pool of routers that you can create to forward traffic. After creating a gateway pool, you can specify the pool as a gateway, within a TMM routing table entry.

**hash persistence**

Hash persistence allows you to create a persistence hash based on an existing iRule.

**header**

A header is supplemental data that an application places before the payload that is being transmitted. A header contains information about how to handle the following payload.
Glossary

health check

A health check is a BIG-IP system feature that determines whether a node is up or down. Health checks are implemented through health monitors. See also health monitor, ECV, EAV, and 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, ECV, and 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 chunking

HTTP chunking refers to the HTTP/1.1 feature known as chunked encoding, which allows HTTP messages to be broken up into several parts. Chunking is most often used by servers when sending responses.

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.

HTTP transformation

When the BIG-IP system performs an HTTP transformation, the system manipulates the Connection header of a server-side HTTP request, to ensure that the connection stays open. See also connection persistence.

ICMP (Internet Control Message Protocol)

ICMP is an Internet communications protocol used to determine information about routes to destination addresses.

i-mode

i-mode® is a service created by NTT DoCoMo, Inc., that allows mobile phone users access to the Internet.

intelligent SNAT

An intelligent SNAT is a SNAT that is implemented by creating a SNAT pool and then writing an iRule that references the SNAT pool. See also SNAT.

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 Security Protocol) 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 BIG-IP system. 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.

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.

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.

Kerberos protocol

The Kerberos protocol is a network authentication protocol that allows individuals communicating over a non-secure network to prove their identity to one another in a secure manner. Kerberos is aimed primarily at a client-server model, providing mutual authentication; both the user and the server verify each other's identity.
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.

LACP (Link Aggregation Control Protocol)

LACP is an industry-standard protocol that aggregates links in a trunk, to increase bandwidth and provide for link failover.

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.

Layer 1 through Layer 7

Layers 1 through 7 refer to the seven layers of the Open System Interconnection (OSI) model. Thus, Layer 2 represents the data-link layer, Layer 3 represents the IP layer, and Layer 4 represents the transport layer (TCP and UDP). Layer 7 represents the application layer, handling traffic such as HTTP and SSL.

Layer 2 forwarding table

A Layer 2 forwarding table correlates MAC addresses of network devices to the BIG-IP system interfaces through which those devices are accessible. On a BIG-IP system, each VLAN has its own Layer 2 forwarding table.

LDAP (Lightweight Directory Access Protocol)

LDAP is an Internet protocol that email programs use to look up contact information from a server.

LDAP authentication module

An LDAP authentication module is a user-created module that you implement on a BIG-IP system to authenticate client traffic using a remote LDAP server.

LDAP client certificate SSL authentication module

An LDAP client certificate SSL authentication module is a user-created module that you implement on a BIG-IP system to authorize client traffic using SSL client credentials and a remote LDAP server.

Least Connections method

Least Connections mode is a dynamic load balancing method that bases connection distribution on which server currently manages the fewest open connections.
link aggregation

Link aggregation is the process of combining multiple links in order to function as though it were a single link with higher bandwidth. Link aggregation occurs when you create a trunk. See also trunk and LACP (Link Aggregation Control Protocol).

Link Aggregation Control Protocol (LACP)

See LACP (Link Aggregation Control Protocol).

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. You can manage local traffic using BIG-IP® Local Traffic Manager.

loopback adapter

A loopback adapter 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.

management interface

The management interface is a special port on the BIG-IP system, used for managing administrative traffic. Named MGMT, the management interface does not forward user application traffic, such as traffic slated for load balancing. See also TMM switch interface.
management route

A management route is a route that forwards traffic through the special management (MGMT) interface.

MCPD service

The Master Control Program Daemon (MCPD) service manages the configuration data on a BIG-IP system.

minimum active members

The minimum active members is the number of members that must be active in a priority group in order for the BIG-IP system to send its requests to that group. If the number of active members falls below this number, requests are sent to the next highest priority group (the priority group with the next lowest priority number).

monitor

The BIG-IP system 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. You can associate monitors with nodes, pools, and individual pool members. See also node, pool, and pool member.

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 BIG-IP system 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.

MSTP (Multiple Spanning Tree Protocol)

Defined by IEEE, MSTP is an enhanced spanning tree protocol. Unlike STP and RSTP, MSTP is VLAN-aware and therefore incorporates the concept of MSTP regions. See also STP (Spanning Tree Protocol) and RSTP (Rapid Spanning Tree Protocol).
MSTP region

An *MSTP region* is a group of Layer 2 devices that have identical values for certain configuration settings. When devices constitute a region, the spanning tree algorithm takes VLANs into account when blocking and unblocking redundant paths.

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 BIG-IP system 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 BIG-IP system does not recognize.

node

A node is a logical object on the BIG-IP system that identifies the IP address of a physical resource on the network. Nodes are directly associated with pool members and monitors. See also *pool member* and *monitor*.

node address

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. See also *pool member*, *monitor*.

node alias

A node alias is a node address that the BIG-IP system uses to verify the status of multiple nodes. When the BIG-IP system uses a node alias to check node status, it pings the node alias. If the BIG-IP system receives a response to the ping, it marks all nodes associated with the node alias as *up*. If the BIG-IP system does not receive a response to the ping, it marks all nodes associated with the node alias as *down*.

node port

A node port is the port number or service name that is hosted by a specific node.
node status

Node status indicates whether a node is **up** and available to receive connections, or **down** and unavailable. The BIG-IP system uses the node ping and health check features to determine node status.

NT LAN Manager

See *NTLM (NT LAN Manager).*

NTLM (NT Lan Manager)

NTLM is an industry-standard technology that uses an encrypted challenge/response protocol to authenticate a user without sending the user's password over the network.

Observed method

Observed method is a dynamic load balancing method that bases connection distribution on the server that currently hosts the fewest connections. Unlike the Least Connections method, the Observed method tracks the number of connections over time and creates a ratio for load balancing.

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 *CRL (certificate revocation list).*

OCSP authentication module

An OCSP authentication module is a user-created module that you implement on a BIG-IP system to authenticate client traffic using a remote OCSP responder. The purpose of an OCSP authentication module is to check on the revocation status of a client SSL certificate.

OCSP responder

An OCSP responder is an external server used for communicating SSL certificate revocation status to an authentication server such as the BIG-IP system.

OCSP responder object

A responder object is a software application on the BIG-IP system that communicates with an OCSP responder, for the purpose of checking revocation status of a client or server SSL certificate.

OneConnect™

The OneConnect feature optimizes the use of network connections by keeping server-side connections open and pooling them for re-use.
optimization profile
An optimization profile is a custom profile that the BIG-IP system provides for you when you install the system. When you use optimization profiles, you ensure the most efficient processing of certain types of HTTP and TCP traffic.

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.

partition
A partition is a logical container that you create, containing a defined set of BIG-IP system objects. You use partitions to control user access to the BIG-IP system. See also user role.

passive monitoring
Passive monitoring occurs as part of a client request. This kind of monitoring checks the health of a pool member based on a specified number of connection attempts or data request attempts that occur within a specified time period. See also active monitoring and simple monitoring.

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/1.1 that allows clients to make requests even when prior requests have not yet received a response from the server.

pool

A pool is a logical group of pool members. The BIG-IP system load balances requests to the pool members within a pool, based on the load balancing method and persistence method you choose when you configure the pool. See also node and pool member.

pool member

A pool member is one of the members of a load balancing pool. A pool member name indicates a node IP address and a service number. See also node.

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 mirroring

Port mirroring is a feature that allows you to copy traffic from any port or set of ports to a single, separate port where a sniffing device is attached.

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 also 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 similar to the Observed method, where nodes are rated according to the number of current connections. However, with the Predictive method, 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.

profile

A profile is a configuration tool containing settings for defining the behavior of network traffic. The BIG-IP system 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 BIG-IP system 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, OneConnect, 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.

RADIUS authentication module

A RADIUS authentication module is a user-created module that you implement on a BIG-IP system to authenticate client traffic using a remote RADIUS server.

RAM cache

A RAM cache is a cache of HTTP objects stored in the BIG-IP system’s RAM that subsequent connections reuse to reduce the amount of load on the back-end servers.

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 Streaming Protocol (RTSP)

See RTSP (Real Time Streaming Protocol).

redundant system

Redundant system refers to a pair of units that are configured for fail-over. In a redundant system, 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.

reference link

A reference link is the lowest-numbered interface in a trunk and is used for link aggregation.

remote administrative IP address

A remote administrative IP address is an IP address from which a BIG-IP system allows shell connections, such as Telnet or SSH.

responder object

See OCSP responder object.

RFC 1918 address

An RFC 1918 address is an address that is within the range of private class IP 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.

router

A router is a Layer 3 networking device. If no VLANs are defined on the network, a router defines a broadcast domain.
RSTP (Rapid Spanning Tree Protocol)
Defined by IEEE, RSTP is an enhanced version of STP (Spanning Tree Protocol). RSTP provides faster spanning tree performance compared to STP. See also STP (Spanning Tree Protocol) and MSTP (Multiple Spanning Tree Protocol).

RTSP (Real Time Streaming Protocol)
RTSP establishes and controls one or more time-synchronized streams of continuous media such as audio or video.

SCTP (Stream Control Transmission Protocol)
SCTP is a general-purpose, industry-standard transport protocol designed for message-oriented applications that transport signalling data.

secure network address translation (SNAT)
See SNAT (secure network address translation).

secure Via header
A secure Via header is data that the BIG-IP system inserts in a SIP message header that indicates where the SIP message originated. See also Via header and header.

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.

server-side SSL profile
A server-side SSL profile is an SSL profile that controls SSL traffic going between a BIG-IP system and a destination server system.

service
Service refers to services such as TCP, UDP, HTTP, and FTP.

services profile
A services profile is a configuration tool on the BIG-IP system for managing either HTTP or FTP network traffic.

Session Initiation Protocol (SIP)
See SIP (Session Initiation Protocol).
**session persistence**
A series of related connections received from the same client, having the same session ID. When persistence is enabled, a BIG-IP system 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 monitoring**
Simple monitoring determines whether the status of a node is **up** or **down**. Simple monitors do not monitor pool members (and therefore, individual protocols, services, or applications on a node), but only the node itself. See also active monitoring and passive monitoring.

**simple persistence**
See source address affinity persistence.

**SIP (Session Initiation Protocol)**
SIP is an application-layer protocol that manages sessions consisting of multiple participants, thus enabling real-time messaging, voice, data, and video.

**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 BIG-IP system. 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 and 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.

source processing
Source processing means that the interface rewrites the source of an incoming packet.

spanning tree
A spanning tree is a logical tree structure of Layer 2 devices on a network, created by a spanning tree protocol algorithm and used for resolving network loops.

spanning tree instance
A spanning tree instance is a specific, named spanning tree that a spanning tree protocol creates. See also spanning tree protocols.

spanning tree protocols
Spanning tree protocols are the IEEE-specified set of protocols known as Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP). The BIG-IP system includes support for all of these protocols. See also STP (Spanning Tree Protocol), RSTP (Rapid Spanning Tree Protocol), and MSTP (Multiple Spanning Tree Protocol).

SSH
SSH is a protocol for secure remote login and other secure network services over a non-secure network.

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 (Secure Network Address Translation) and intelligent SNAT.
standby unit
A standby unit in a redundant system is a unit that is always prepared to become the active unit if the active unit fails.

state mirroring
State mirroring is a feature on the BIG-IP system that preserves connection and persistence information in a redundant system.

static route
A static route is a route that you must explicitly configure on a Layer 3 device in its routing table. See also dynamic route.

static self IP address
A static self IP address is a self IP address that is not shared between two units of a redundant system.

sticky persistence
See destination address affinity persistence.

STP (Spanning Tree Protocol)
Defined by IEEE, STP is a protocol that provides loop resolution in configurations where one or more external switches are connected in parallel with the BIG-IP system. See also RSTP (Rapid Spanning Tree Protocol) and MSTP (Multiple Spanning Tree Protocol).

Stream Control Transmission Protocol (SCTP)
See SCTP (Stream Control Transmission Protocol).

subdomain
A subdomain is a sub-section of a higher level domain. For example, .com is a high level domain, and F5.com is a subdomain within the .com domain.

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.
TACACS+ authentication module
A TACACS+ authentication module is a user-created module that you implement on a BIG-IP system to authenticate client traffic using a remote TACACS+ server.

tagged interface
A tagged interface is an interface that you assign to a VLAN in a way that causes the system to add a VLAN tag into the header of any frame passing through that interface. Tagged interfaces are used when you want to assign a single interface to multiple VLANs. See also VLAN (virtual local area network).

Tcl
Tcl (Tools Command Language) is an industry-standard scripting language. On the BIG-IP system, users use Tcl to write iRules.

TMM (Traffic Management Microkernel) service
The TMM service is the process running on the BIG-IP system that performs most traffic management for the product.

TMM switch interface
A TMM switch interface is an interface that the BIG-IP system uses to forward user application traffic such as HTTP or SSL traffic. Thus, when load balancing application traffic, the BIG-IP system uses TMM switch interfaces. See also management interface.

TMM switch route
A Traffic Management Microkernel (TMM) switch route is a route that forwards traffic through the TMM switch interfaces and not the management interface.

traffic class
A traffic class allows you to classify traffic according to a set of criteria that you define, such as source and destination IP addresses. With a traffic class, the BIG-IP system can regulate the flow of traffic based on that classification.

transparent node
A transparent node appears as a router to other network devices, including the BIG-IP system.

trunk
A trunk is a combination of two or more interfaces and cables configured as one link.
trusted CA file

A trusted CA file is a file containing a list of certificate authorities that an authenticating system can trust when processing client requests for authentication. A trusted CA file resides on the authenticating system and is used for authenticating SSL network traffic.

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.

Universal Inspection Engine (UIE)

The Universal Inspection Engine (UIE) is a feature that offers universal persistence and universal content switching, to enhance your load balancing capabilities. The UIE contains a set of rule variables and functions for building expressions that you can specify in pool definitions and rules.

universal persistence

Universal persistence gives you the ability to persist on any string found within a packet. Also, you can directly select the pool member to which you want to persist.

user configuration set (UCS)

A user configuration set is a backup file that you create for the BIG-IP system configuration data. When you create a UCS, the BIG-IP system assigns a .ucs extension to the filename. See also archive.

user role

A user role is a type and level of access that you assign to a BIG-IP system user account. By assigning user roles, you can control the extent to which BIG-IP system administrators can view or modify the BIG-IP system configuration.

Via header

A Via header is data that the BIG-IP inserts in a SIP message that indicates where the SIP message originated. See also secure Via header and header.

virtual address

A virtual address is an IP address associated with one or more virtual servers managed by the BIG-IP system. See also virtual server.

virtual port

A virtual port is the port number or service name associated with one or more virtual servers managed by the BIG-IP system. 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 BIG-IP system or other type of host server.

VLAN (virtual local area network)

A VLAN is a logical grouping of interfaces connected to network devices. You can use a VLAN to logically group devices that are on different network segments. Devices within a VLAN use Layer 2 networking to communicate and define a broadcast domain.

VLAN group

A VLAN group is two or more VLANs that you put together into a VLAN group. A primary use of a VLAN group is to successfully route traffic when both the source and the destination hosts reside on the same network.

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 (virtual local area network).

VLAN tag

An IEEE standard, a VLAN tag is an identification number inserted into the header of a frame that indicates the VLAN to which the destination device belongs. VLAN tags are used when a single interface forwards traffic for multiple VLANs.

WAP (Wireless Application Protocol)

WAP is an application environment and set of communication protocols for wireless devices designed to enable manufacturer-, vendor-, and technology-independent access to the Internet and advanced telephony services.

watchdog timer card

A watchdog timer card is a hardware device that monitors the BIG-IP system for hardware failure.

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.
WKS (well-known services)

Well-known services are protocols on ports 0 through 1023 that are widely used for certain types of data. Some examples of some well-known services (and their corresponding ports) are: HTTP (port 80), HTTPS (port 443), and FTP (port 20).
Index
Index

/config/eav file A-47
/etc/bigip.conf file 17-17

A
A 15-1
acceleration
and CMP 2-8
and PVA 8-5
See hardware acceleration.
Accept-Encoding headers 6-12
Accounting Bug setting 10-12
accounting, RADIUS A-69
Active Directory servers, for authentication 10-4
adaptive compression
configuring on 6400, 6800, and 8400 6-26, 6-27
introducing 6-24
additional information
for Configuration Worksheet 1-23
for Platform Guide 1-23
for TMOS Management Guide for BIG-IP Systems 1-23
address data groups 17-15
address translation 1-7, 13-1
address translation, standard 14-2, 14-3
address translation setting 2-18
admin account 10-5
administrative partitions
and application templates 1-19
agent types B-10
AIA field 10-31
alert timeouts, specifying 9-22
anonymous searches, allowing 10-5
Apache variants 7-9, 7-10
application deployment guides 1-20
application security classes 8-14
Application Security Manager 8-14
application templates
described 1-19
using 1-20, 1-21
viewing and deleting 1-21
application traffic
authenticating 10-1
configuring 1-19
managing 5-1
ARP requests 2-21
ASN.1 DER format C-3
authentication depth 9-28
authentication iRules
assigning 10-9, 10-15, 10-46
defined 10-2
authentication methods 9-26
authentication module types 10-1
authentication modules
defined and listed 10-1
implementing 10-2
listed 1-5
authentication profiles
assigning 10-9, 10-15, 10-46
defined 10-2
listed 5-2
authentication, per session 9-27
AuthorityInfoAccess field
See AIA field.
authorization
and groups/roles 10-23
using SSL certificates 9-3
authorization failures 6-4
authorization options, listed 9-6
authorization parameters, listed 10-24, 10-36
automatic encoding 7-10

B
bandwidth
borrowing 16-8, 16-9
replenishing 16-6
saving 16-6
base packet rates 16-6, 16-9
Base Rate setting, configuring 16-5
base throughput rate 16-5
BEA WebLogic application 1-19
BIG-IP object status
displaying 1-17
BIGIpCookie cookie name 7-9
BIGipServer<pool_name> cookie name 7-8, 7-9
bigpipe -? command 17-5, 17-10
Bind DN setting 10-5
Bind Password setting 10-5
Bind Time Limit setting 10-5
blank cookies
about 7-9
inserting and searching for 7-10
browser limits
and network map 1-15
browser workarounds 6-22
browsers
and data compression 6-12
and shutdown alerts 9-23
and trusted CAs 9-28
buffer size, for compression 6-19
burst reservoirs 16-6
Burst Size setting, configuring 16-6
bursting restrictions 16-7
Index

C

Cache items
- clearing 6-33
- listed 6-30
Caching profiles 6-36
Caching proxy servers 7-11
CAs 10-31
Ceiling Rate setting, configuring 16-5
Ceiling rates 16-6, 16-9
Certificate archives, creating 9-11
Certificate authentication features 9-2
Certificate authorities
- See CAs.
Certificate chain files 9-28
Certificate comparison search method 10-22
Certificate information, viewing C-4
Certificate issuers 9-7
Certificate mapping search method 10-22
Certificate request methods 9-8
Certificate requests, transmitting 9-8
Certificate revocation
- See CRLs.
Certificate Revocation List Distribution Points
- See CRLDP module.
Certificate revocation lists
- See CRLs.
Certificate revocation status
- assessing 10-30, 10-31
- checking 10-31
Certificate status, displaying 9-7
Certificate verification C-4
Certificate verification failures 9-28, 10-32
Certificate verification, client-side 9-4
Certificate verification, server-side 9-4
Certificate/key pairs, requesting 9-8
Certificates
- deleting 9-10
- for authorization 10-21
- generating and installing 9-4
- importing 9-10
- inserting as headers 9-6
- managing 9-7
- purpose of 9-3
- renewing 9-9
- requesting from CAs 9-8
- requiring 9-26
- trusting 9-15
Certificate, revoking C-3
Certificate verification, client-side 9-16
Certificate verification, server-side 9-16
Certificate chain files 9-4, 9-15
Check SSL Peer setting 10-6
Child rate classes 16-8
Chunking 6-7
Ciphers, specifying in headers 9-16, 9-20
Client authentication methods 9-26
Client certificates, creating C-1
Client certificates, requiring 9-26
Client connections, authenticating 9-26, 9-30
Client CRL files, specifying 9-29
Client CRL paths, specifying 9-29
Client headers, for authorization 9-6
Client ID setting 10-12
Client IP addresses
- inserting into headers 6-5
- preserving 6-5
- tracking connections for 7-20
Client request, authorizing 10-21
Client traffic
- and rate classes 16-8
- directing for a subnet 2-5
- redirecting 4-1
Client Trusted CAs file, described 9-16
Client Trusted CAs path, described 9-16
Client verification process 9-4
Clients, and secure connections 6-8
Client-side connections
- and certificate verification 9-16
- and trusted CAs 9-4
Client-side iRule context 17-9
Client-side profiles, and encryption/decryption 9-5
Client-side session cache size 9-21
Client-side SSL profiles 9-1
Clone pool, and virtual servers 2-18
Clone pools 2-18
Clustered multi-processing
- See CMP.
CMP 2-8
Compression
- and CPU usage 6-22
- and RAM Cache feature 6-30
Compression buffer size 6-19
Compression profiles 6-36
Compression providers, understanding 6-23
Compression settings 6-14, 6-34
Compression statistics, described 6-29
Compression strategies
- described 6-24
- understanding 6-24
Compression, enabling 6-16
Concurrent connection limits 3-7, 4-17
Concurrent connections 2-21
Configuration settings 10-7, 10-13, 10-44
Configuration utility
- about online help 1-26
- and Welcome screen 1-26
- for Setup utility 1-7
- purpose 1-7
- using web-based 1-7
Configuration Worksheet 1-23
Confirm Bind Password setting 10-5
Confirm Secret setting 10-11, 10-41
Index

connection distribution 4-14
connection limits
  allowing 2-21
  for CMP-enabled systems 2-9
  for nodes 3-7, 4-17
connection persistence 6-7
connection pooling
  defined 1-4, 11-2
  for NTLM
  for NTLM connections 11-4
  See also X-Forwarded-For header.
connection queueing 16-9
connection requests
  receiving 2-5
connection termination 9-27
connections
  and certificate verification 9-16
  and shutdown alerts 9-23
  and trusted CAs 9-4
  authenticating 10-4, 10-10
  distributing by priority 4-14
contains operator 17-14
content
  examining 8-11
content compression 6-30
content searching 17-1
content switching
  customizing 17-1
  defined 1-3
Content-Encoding headers 6-12
Content-Type responses 6-5
context, for iRules 17-9
conventions 1-24
cookie encryption and decryption 6-9
Cookie Hash mode 7-10
cookie names 7-8
cookie persistence
  defined 7-7
  See also HTTP cookie persistence.
cookie profile settings 7-7
cookie templates 7-10
cookie values 7-10
cookies
  See also HTTP cookie persistence.
cookies, and HTTP Cookie Rewrite 7-9
CPU metrics, gathering A-55, B-10
CPU usage
  and compression 6-22
criteria specification 8-11
CRL distribution points 10-40
CRL files 9-29
CRLDP authentication module C-3
CRLDP module, defined 10-40
CRLs
  and LDAP servers 10-40
  bypassing 10-31
creating C-3
  described 9-5
drawbacks 10-30
for client-side SSL 9-29
generating
  See also OCSP.
timeout of 10-43
  viewing C-4
custom HTTP profiles 5-6
custom LDAP profiles 10-8
custom monitors 12-5
custom profiles 5-4, 5-5
D
data collection agents B-10
data compression
  described 6-12
  enabling 6-16
data compression profiles 6-36
data group members 17-19
data group size 17-15
data group storage
  See also in-line data group storage.
  See external data group storage.
data group types 17-15
data groups
  configuring 17-14
  storing 17-18
Debug Logging setting 10-6, 10-12
decryption
  described 9-4
  of cookies 6-9
  summary of 9-3
default compression values 6-14, 6-34
default gateway 14-2
default gateways 14-2, 14-3
  and addresses translation 2-5
default HTTP profile 5-6, 6-2
default LDAP profile, modifying 10-8
default persistence profiles 2-20
default profiles 5-3
  summarized 5-5
  using 5-3
default RADIUS profile 10-44
default routes 14-2
default wildcard virtual servers, creating 2-12
deflate compression method 6-18
destaddr profile settings 7-11, 7-13
destination address affinity persistence 7-11
destination address ranges 2-5
destination address ranges, directing to 2-5
destination IP addresses
  and persistence 7-11
  translating 2-3, 13-1
destination IP translation 2-3, 14-3
Index

Direction setting, configuring 16-8
dirname-based addresses C-1
disk metrics, gathering A-55, B-10
distinguished names 10-5
distinguished names, specifying 10-5
DNS lookups A-24
dynamic IP addresses, and persistence 7-21
Dynamic Ratio mode
  configuring RealSystem Servers for B-1
  configuring WMI for B-3
described 4-12

E
embedded distribution points C-1
encoding, chunked and unchunked 6-7
encryption
  described 9-5
  of cookies 6-9
  summary of 9-3
ephemeral ports 14-3
equations, and encoding 7-10
error codes 6-5
event declarations 17-8
event execution, terminating 17-10
event-based traffic management 17-8
expired session IDs 9-21
external data group storage 17-15
external data groups 17-18

F
fallback hosts 6-5
fallback persistence profiles 2-20
Fast HTTP profile settings 8-7
Fast L4 profile settings 8-2
Fastest mode, described 4-12
file access
  and NTLM
files, including and excluding 6-13
Filter setting 10-6
firewalls 2-6
floating IP addresses
  and SNATs 14-5
flooding
  resistance to 8-25
format
  See PEM format.
format strings 17-5
forwarding virtual servers 2-2
FQDNs, and redirection 6-5
FTP profile settings 6-40

G
genconf and genkey utilities 9-4
Group DN setting 10-6
Group Member Attribute setting 10-6
group-based authorization 10-23
group-based LDAP authorization 10-23
gzip compression method 6-18
gzip compression, and memory levels 6-20

H
hardware acceleration 8-2, 8-5
hash persistence, defined 7-12
header content, erasing 6-6
header insertion
  and client-side authentication 9-27
  for cookie persistence 7-8
header insertion syntax 6-5, 6-6
header searching 17-1
headers
  examining 8-11
  for client authorization 9-6
  inserting 6-6
health monitors
  configuring B-10
  for pools 4-7
  logical grouping in 4-8, 4-18
  transparent mode in 4-8, 4-18
help, online 1-26
high-demand objects 6-30
host IP address data groups 17-15
host names, redirecting 6-5
Host setting, for RADIUS objects 10-10
host virtual servers
  defined 2-5
Hosts setting 10-5
HTTP authorization failures 6-4
HTTP Class profile settings 8-11
HTTP compression settings 6-14, 6-34
HTTP content
  examining 6-5
HTTP Cookie Insert method 7-8
HTTP Cookie Passive mode 7-10
HTTP cookie persistence 7-7, 7-8
HTTP Cookie Rewrite method 7-9
HTTP data compression
  described 6-12
  enabling 6-16
HTTP data compression profiles 6-36
HTTP header content, erasing 6-6
HTTP headers
  examining 6-5
  inserting 6-5
HTTP Location header 6-8
HTTP methods, caching 6-31
HTTP monitors A-66
HTTP objects
  storing 6-30
HTTP optimization profiles 6-36
HTTP pipelining
  See pipelining.
HTTP profile settings, configuring  6-2
HTTP profile sharing  1-21
HTTP profiles
  default and custom  5-6
  described  6-1
HTTP Redirect Rewrite setting  6-5
HTTP redirections
  and pool selection  17-5
  example of  17-5
  rewriting  6-8
HTTP requests, redirecting  17-5
HTTP responses, compressing  6-13
HTTP rewrites, examples of  6-9
HTTP traffic management  6-1
HTTP/1.0 compression  6-21
httpd.conf file, and cookies  7-9, 7-10
I
IDEA cipher suite  9-17
identities, trusting  9-6
Idle Timeout setting
  for LDAP profiles  10-8
  for RADIUS profiles  10-14, 10-44
Ignore AIA parameter  10-31
ignore option  9-26
Ignore Unknown User setting  10-6
IIS application  1-19
imap monitors  A-25
inbound connections
  See incoming connections.
inbound SNAT
  defined  14-3
inbound SNATs
  and default gateways  14-2
  and OneConnect  14-2
  for same network  14-2
  illustrated  14-4
incoming connections
  and NATs  13-2
in-line data group storage  17-17
Insert mode, for HTTP cookie persistence  7-8
integer data groups  17-16
intelligent SNATs  14-9
intermediate CAs, trusting  9-15
intrusion detection  2-18
invalid protocol versions, configuring  9-20
IP address data groups  17-15
IP address destinations  2-6
IP addresses
  and persistence  7-21
  and redirection  6-5
  and virtual servers  2-5
  for clients  7-20
in cookies  7-9
mapping to SNAT pools  14-6
matching  2-5, 2-6
sharing  2-2
  specifying for NATs  13-6
  translating  2-6
iRule command types  17-3
iRule elements  17-2
iRule evaluation, controlling  17-8
iRule event declarations  17-2
iRule event types  17-9
iRule examples
  HTTP redirection iRule  17-5
iRule operators  17-3
iRule prerequisites  17-8
iRule setting
  for LDAP profile  10-8
  for RADIUS profiles  10-14, 10-44
iRules
  and ciphers  9-5
  and HTTP header insertion  9-6
  and network map  1-18
  and SNATs  14-6
  and virtual servers  17-7
  assigning  2-24, 17-10
  associating with virtual servers  17-7
  creating  17-7
  defined  17-1
  for authentication  10-2
  for SSL traffic management  9-1
iRules statement commands,  17-11
Issuer field  10-31
K
Keep-Alive support, adding  1-4
kernel routing table  2-21
key archives, creating  9-11
key encryption  9-15
key pairs, importing  9-10
key types  9-7
key/certificate archives, importing  9-10
key/certificate pairs, importing  9-4
keys
  deleting  9-10
  managing  9-7
  keys, generating  C-1
  keywords, reserved  12-9, A-1
L
L2 forwarding virtual servers, defined  2-2
L3 Address mode  4-14
last hop pools, and virtual servers  2-18
LDAP authorization concepts  10-22
LDAP authorization criteria  10-23
LDAP authorization parameters, listed  10-24, 10-36
LDAP base directory name 10-41
LDAP configuration objects, creating 10-5
LDAP CRL distribution point C-2
LDAP database, searching 10-22
ldap default profile, modifying 10-7, 10-8
LDAP module, defined 10-1
LDAP monitors A-23
ldap pre-configured monitor A-23
LDAP profiles
  and virtual servers 10-9
  creating 10-7
LDAP servers 10-40
  and traffic authentication 10-21
    for authentication and authorization 10-4
    for traffic authentication and authorization 10-16,
    10-21, 10-47
LDAP Version setting 10-5
Least Connections mode, described 4-13
Least Sessions mode 4-14
Lightweight Directory Access Protocol module
  See LDAP module, defined.
linear white space, managing 6-10
load balancing methods
  and CMP 2-9
  default 4-11
  defined 4-1
load balancing pools 1-12, 4-1
load balancing virtual servers 2-1
load balancing, introduced 1-7
local traffic management, defined 1-1
local traffic summary 1-15, 1-17
Location header 6-8
logical operators, listed 17-3
Login Attribute setting 10-6

M
man-in-the-middle attacks, preventing 9-22
manual resume attribute 3-9, 12-12
map
  for viewing network 1-17
masks, for simple persistence 7-20
masquerade attacks
  resistance to 8-25
matchclass command 17-14
MD5 hash 9-21
memory levels, for gzip compression 6-20
memory metrics, gathering A-55, B-10
meta-data, for external data groups 17-17
methods, caching 6-31
Microsoft Exchange Outlook Web Access application 1-19
Microsoft IIS application 1-19
Microsoft SharePoint application
  SharePoint application 1-19
min_active_members value 4-14
minimum health monitors 3-7
ModSSL method emulation, enabling and disabling 9-20
monitor association types 12-15
monitor instances, enabling and disabling 12-17
monitor settings 12-4
monitor-pool associations, managing 4-28
  for nodes 3-5
  for pools 4-7
  managing 12-17
MSRDP persistence
  and older platforms 7-15
    enabling 7-14, 7-15
MSRDP platform requirements 7-15
MSRDP profile settings 7-16
multiple requests
  initiating 6-10

N
Nagle’s algorithm 8-20, 8-22, 8-23
NAT addresses
  sharing 13-3
NAT connections
  tracking 13-3, 14-1
NAT settings 13-5, 13-6
NATs
  and open ports 13-2
    configuring 13-6
    defined 13-1
    examples of 13-3, 13-4
  netmasks, specifying 2-12
  network IP address data groups 17-15
  network map
    defined 1-17
    displaying 1-17, 1-18
  network map feature
    defined 1-14
  Network maps
    for application templates 1-22
    network performance, optimizing 1-4
    network traffic
      authenticating 10-1
      managing 5-1
    network translation addresses (NATs)
      See NATs.
  network virtual server types 2-5
network virtual servers
  defined 2-5
ntmp monitors A-27
node availability 3-9, 12-12
node information, in cookies 7-10
nodes
  and connection limits 3-7, 4-17
  as pool members 4-7
defined 3-1
directing traffic to 7-21
receiving connections 4-11
NT Lan Manager
defined 11-4
See also NTLM.
See NTLM profiles
NTLM profile
purpose of 11-4
NTLM profiles 2-16
numeric value classes 17-16

O
object status 1-16
Observed mode, described 4-13
OCSP configuration objects, creating 10-33, 10-36
OCSP module 10-2
OCSP prerequisites 10-32
OCSP profiles
and virtual servers 10-39
creating 10-37
OCSP protocol
defined 10-31
described 10-30
OCSP responder definitions, choosing 10-31
OCSP responder objects
creating 10-33
defined 10-23
OCSP responders
and CAs 10-32
choosing 10-31
OneConnect
and inbound SNATs 14-2
OneConnect profile settings 11-2
OneConnect, enabling 6-7
Online Certificate Status Protocol module
See OCSP module.
online help 1-26
OpenSSL 0.9.8.x  C-1
openssl command 9-29
openssl utility  C-1, C-2, C-3, C-4
OpenSSL web site 9-17
operators 17-3
optimization profiles 6-36
Oracle Application Server application 1-19
order, of packets 16-1
outbound connections
See outgoing connections.
outbound traffic, and ToS level 4-10
outgoing connections
and NATs 13-4
Outlook Web Access application 1-19

P
packets, queuing and dequeuing 16-1, 16-9
PAM authentication modules, listed 1-5
PAM modules 10-1
PAM, defined 10-1
parameters, for LDAP authorization 10-21
Parent Class setting, configuring 16-8
parent HTTP profiles, specifying 6-4
Parent Profile setting 10-7
parent profiles
defined 5-5
parent rate classes, borrowing bandwidth from 16-8
parent SIP profiles, specifying 6-44
partitions
and data groups 17-14
and iRules 17-7
and monitors 12-8, 12-17
and nodes 3-2
and pools 4-2
and profiles 5-8
and rate classes 16-2
and SNAT pools 14-7, 14-10
and SNATs 14-10
and virtual servers 2-11
pass phrase encryption 9-15
pass phrase length 9-15
pass phrases
See SSL pass phrases.
passive mode 7-10
passwords, adding and stripping  C-4
pattern strings
for traffic classification 6-12, 8-11
peer authentication 9-16
PEM format 9-10, 9-15
PEM format conversion  C-4
performance
and load balancing 4-14
performance (HTTP) virtual servers 2-2
performance (layer 4) virtual servers 2-2
persistance
and conditions for 7-5
and iRules 17-1, 17-13
and MSRDP platform requirements 7-15
and need for 7-1
and plain-text traffic 7-21
for SSL connections 9-6
See also connection persistence.
See also session persistence.
persistence profile types 7-4
Index

persistence profiles
   and iRules 7-2
   listed 5-1, 5-2
persistence timer 7-20
PFIFO, defined 16-10
pipelining
   defined
       HTTP pipelining
       defined 6-10
pipelining
   defined 1-4
PKCS12 file, creating C-2
plain-text traffic, load balancing 7-21
Platform Guide 1-23
platform requirements, for MSRDp persistence 7-15
Pluggable Authentication Modules
   See PAM modules.
   See PAM.
pool member availability 12-12
pool member status 4-25
pool members
   adding 4-15
   as servers 4-7
   defined 1-12, 4-1
   selecting with iRules 17-4
pool monitoring 4-7
pool naming 4-7, 14-11
pool screen, shown 1-12
pool settings, and default values 4-16
pool status 4-22
pool-monitor associations, managing 4-28
pools
   and SNAT/NAT connections 4-9
   defined 4-1
   deleting 4-22, 4-27
   selecting with iRules 17-1, 17-4
pop3 monitors A-35
port numbers
   in cookies 7-9
   redirecting 6-5
   rewriting 6-8
port translation
   for NATs 13-2
port translation setting 2-18
port translation, turning off 2-12, 2-13
port-specific wildcard virtual servers, creating 2-12, 2-13
Predictive mode, described 4-13
Priority FIFO
   See PFIFO.
priority member activation 4-14
priority numbers, assigning 4-14
Privacy Enhanced Mail format
   See PEM format.
private class IP addresses
   mapping to public 13-1
profile dependencies 5-13
profile names, specifying 6-4, 6-44
profile settings, overriding 9-12, 17-12
profile summary 5-5
profile types 5-1
profiles
   and virtual servers 2-15, 5-12
   associating with virtual server 1-12
   creating custom 5-4
   creating parent 5-4
   defined 1-12, 5-1
   described 5-1, 9-11
   using default 5-3
protocol names, rewriting 6-8, 6-9
protocol profiles
   listed 5-1, 5-2
Protocol Security Manager
   See PSM.
protocol versions
   configuring 9-20
   specifying 9-16, 9-20
protocols
   and persistence settings 7-18, 7-20
   and virtual servers 2-15
proxy servers 2-6
PSM
   and SMTP profiles
public IP addresses
   mapping to private 13-1
PVA hardware acceleration 8-2, 8-5
Q
QoS level
   setting 4-10
QoS pool attribute 4-10
query commands, defined 17-3
Queue Discipline setting, configuring 16-9
R
RADIUS accounting A-69
RADIUS configuration objects, creating 10-11, 10-42
RADIUS module, defined 10-1
RADIUS monitors A-37
RADIUS profiles
   and virtual servers 10-15, 10-46
   creating 10-13, 10-43
   modifying 10-44
RADIUS server objects, creating 10-10, 10-41
RADIUS servers
   and authentication 10-10
   and traffic authentication 10-10
RAM Cache feature 6-30
RAM Cache specifications 6-30
rate class example 16-7
rate class settings 16-4
rate classes
  and direction 16-8
  assigning 16-2
  creating 16-2
  defined 16-1, 16-2
  managing 16-11
  naming 16-5
rate shaping, defined 1-4, 16-1
rate, of packets 16-1
Ratio method, described 4-12
ratio weights, specifying 4-17
Real Time Streaming Protocol
  See RTSP profiles
Real Time Streaming Protocol (RTSP)
  See RTSP protocol.
RealSystem Servers, configuring for load balancing 8-1
Record-Route header, inserting into a SIP message 6-45
redirect iRule command 17-5
redirection
  and pool selection 17-5
  defined 6-5
  rewriting 6-8
redirection rewrites
  enabling 6-8
  examples of 6-9
regular expressions
  for data compression 6-13
  for traffic classification 6-12, 8-11
relational operators, listed 17-3
remote authentication
  and TMM service 10-1
Remote LDAP Tree setting 10-5
request option 9-26
requests, chunking and unchunking of 6-7
require option 9-26
reserved keywords 12-9, A-1
reservoirs
  See burst reservoirs.
resources, controlling 9-6
Responder CAs parameter 10-32
responder definitions, choosing 10-31
responder objects, defined 10-2, 10-23
Responder URL parameter 10-31
responders
  and CAs 10-32
  choosing 10-31
response acceptance 14-4
response rejection 2-3, 14-2
  illustrated 14-3
responses
  chunking and unchunking 6-7
  compressing 6-13
Retries setting 10-12
return path 14-2
Reverse setting 12-12
reverse translation 2-3
revocation
  See CRLs.
revocation status
  assessing 10-30
  checking 10-31
revocation, of certificates 9-5
Rewrite mode 7-9
role-based authorization 10-23
role-based LDAP authorization 10-23
Round Robin mode, described 4-11
route advertisement 2-21
route insertion 2-21
routers 2-6
routing problems 14-3
RPC monitor A-39, A-44
RSA keys C-4
RTCP Port 6-48
RTP Port 6-48
RTSP profile configuration 6-48
RTSP profile settings 6-47, 6-50
RTSP profiles 2-17
  and TCP 6-47
  and virtual servers 6-47, 6-48
RTSP protocol
  defined 6-47, 6-50
RTSP tunneling 6-47
rule operators, listed 17-3
rules
  See iRules.
S
SAP application 1-19
SASP monitors A-44
Scripted monitor A-46
SCTP profile settings 8-25
SCTP protocol
  defined 8-25
SCTP streams 8-25
search account 10-5
Search box
  for filtering profile list 5-11
  for network map 1-14
search methods, for LDAP database 10-22
Search Time Limit setting 10-5
Secret setting 10-11
secure network address translation
  See SNATs.
Secure Via header, inserting into a SIP message 6-45
security breaches, preventing 1-5
See also outbound SNATs
self IP addresses
  and SNATs 14-5
self-IP addresses, assigning 2-12
self-signed certificates, generating and requesting 9-8
server authentication C-2
Index

server availability 3-9, 12-12
server availability, increasing 2-1
server certificates, creating C-2
server chain files, described 9-4
server load 6-30
server nodes
  and default routes 14-2
server objects, defined 10-2
server overload 4-1
server traffic, and rate classes 16-8
servers, selecting with iRules 17-4
server-side connections 1-4
server-side iRule context 17-9
server-side session cache size 9-21
server-side SSL connections
  and certificate verification 9-16
  and trusted CAs 9-4
server-side SSL profiles
  and encryption 9-5
  defined 9-1
  managing 9-2
server-side verification, described 9-4
service checks, troubleshooting A-29
Service Port setting
  for LDAP modules 10-5
  for RADIUS objects 10-10, 10-41
services profiles
  listed 5-1
services profiles, listed 5-1
session authentication 9-27
session cache size 9-21
session cache timeout 9-21
Session Directory service 7-15
Session Directory, and MSRDp persistence 7-16
session IDs, inserting 9-6
Session Initiation Protocol 6-42, 7-18
session loss
  monitoring and detecting 8-25
session persistence
  and iRules 17-13
  enabling 2-1
  for SSL connections 9-6
session renegotiation, forcing 9-22
session sharing 7-16
session termination criteria, enabling for SIP profiles 6-44
settings
  for Protocol profiles 8-1
SFQ queueing method
  defined 16-9
shutdown alerts 9-23
signed certificates
  for authorization 10-21
  See also certificates.
Simple Mail Transport Protocol
  See SMTP profiles
simple persistence
  See source address affinity persistence.
SIP header, handling messages in 6-45
SIP header, specifying size of 6-44
SIP monitor A-48
SIP persistence 6-46, 7-18
SIP persistence, defined 7-18
SIP profile settings 6-42, 7-18
SIP profile settings, configuring 6-42
size, of SSL session cache 9-21
SMB monitor A-51
SMTP monitors A-53
SMTP profiles 2-17
SNAT
  examples of 14-4
SNAT addresses
  and default gateways 14-2
SNAT automap
  defined 14-5
SNAT behavior 14-4
SNAT connections
  tracking 14-1
SNAT pool setting 2-18
SNAT pool settings 14-7
SNAT pools
  and partitions 14-7
  and virtual servers 2-18
  assigning to virtual server 14-13
  creating 14-8
  defined 14-6
snatpool command 14-9, 17-6
SNATs
  and default gateways 14-2
  and OneConnect 14-2
  and virtual bounceback 14-1
  examples of 14-18
  See also inbound SNATs
SNATs, enabling and disabling 4-9
SNATs-to-NATs comparison 14-1
SNMP DCA Base monitor B-10
SNMP DCA monitor B-10
snmp_dca_base template A-57
SOAP monitor A-58
source address affinity persistence 7-20
source address affinity profile settings 7-20
source IP addresses
  translating 13-1
source IP translation 2-3, 14-3
SQL Enterprise Manager A-29
SQL-based service checks, troubleshooting A-29
SQL-based services, and service checks A-27
SSL authentication
  and certificate revocation 10-30, 10-40
SSL authentication features 9-2, 9-25
SSL authorization
  described 9-3
SSL CA Certificate setting 10-6
SSL Certificates screen 9-7
SSL certificates, managing 9-7
SSL Ciphers setting 10-6
SSL client certificate LDAP configuration objects 10-24
SSL client certificate LDAP module, defined 10-2
SSL client certificate LDAP profiles 10-27, 10-29
SSL Client Certificate setting 10-6
SSL Client Key setting 10-6
SSL configuration tasks 9-1
SSL connections, and shutdown alerts 9-23
SSL defect workarounds, configuring 9-17
SSL encryption/decryption 9-5
SSL feature summary 9-2
SSL key encryption 9-15
SSL keys, managing 9-7
SSL OCSP configuration objects, creating 10-36
SSL pass phrases
  encrypting 9-15
  for key encryption 9-15
SSL persistence profile settings 7-22
SSL persistence types 9-6
SSL persistence, defined 7-21
SSL profile names, specifying 9-14
SSL profile types 9-1, 9-12
SSL profiles
  listed 5-2
SSL profiles, defined and listed 5-1, 9-1
SSL protocol versions, configuring 9-20
SSL session cache size 9-21
SSL session cache timeout 9-21
SSL session renegotiation, forcing 9-22
SSL sessions, negotiating 9-28
SSL setting 10-6
SSL shutdowns 9-23
SSL timeout duration 9-22
SSL verification, and chain files 9-4
SSL, authenticating to clients 9-15
SSLv2 protocol 9-20
SSLv3 protocol 9-20
standard SNATs 14-9
statement commands
  defined 17-3
  specifying 17-11
static content 6-30
statistics
  displaying 2-8
Statistics profile 11-6
statistics, for virtual servers 2-24
status icons
  for nodes 3-9
  for pool members 4-25
  for pools 4-22
  for virtual servers and addresses 2-26
sticky persistence
  See destination address affinity persistence.
sticky persistence type 7-11
Stochastic Fair Queueing
  See SFQ queueing method.
Stream Control Transmission Protocol (SCTP)
  See SCTP protocol.
Stream profile 11-7
streaming media protocols
streaming-media presentations
string data groups 17-16
string matching 6-12, 8-11
stylistic conventions 1-24
subnets
  for server nodes 14-2
SYN flooding, preventing 1-5
SYSLOG debugging 10-6
system performance, monitoring of 1-7
T
TACACS+ configuration objects, creating 10-16, 10-48
TACACS+ module, defined 10-2
TACACS+ profiles
  and virtual servers 10-20, 10-51
  creating 10-18, 10-49
TACACS+ servers, and traffic authentication 10-16
Tcl syntax 17-2
TCP connections, and shutdown alerts 9-23
TCP monitors A-66
TCP streams 8-25
templates
  for application configuration 1-19
  See also application templates
Terminal Server configuration 7-15
test accounts, creating A-29
throughput
  customizing 1-4
  enforcing 16-1
throughput limitations 16-1, 16-2
throughput policies 16-1
throughput policy, enforcing 16-1
throughput rates 16-5
throughput restrictions, applying 16-8
Timeout setting, for RADIUS objects 10-11
timeout values 7-20
TLSv1 protocol 9-20
TMM instances, multiple 2-8
TMM service
  and remote authentication 10-1
TMOS Management Guide for BIG-IP Systems 1-23
Tools Command Language syntax 17-2
ToS field
  and queueing 16-10
ToS level, setting 4-10
ToS pool attribute 4-10
traffic
  and QoS level 4-10
  authenticating 10-1
  distributing by priority 4-14
  managing 5-1
  queueing 16-10
  redirecting 6-5
traffic acceleration
  and CMP 2-8
  and PVA 8-5
  using Fast L4 profile 8-2
traffic class list 2-17
traffic class settings 15-2
traffic classes
  creating 15-1
  defined 15-1
  managing 15-3
traffic classification 8-11
traffic direction, and rate classes 16-1
traffic flow limits 16-5
traffic flow rates 16-6, 16-9
traffic flow tagging 2-17
traffic queueing 16-9
traffic rates, bursting 16-6
traffic replication 2-18
traffic types, managing 2-1
traffic, copying 2-18, 2-19
translation address properties 14-11
translation addresses
  and persistence 7-21
  choosing for SNATs 14-9
transparent device pools, creating 2-12
transparent devices, receiving connections from 2-7
transparent mode 4-8, 4-18
transparent nodes 2-6, 2-7
Transparent setting 12-11
trusted CA file names, specifying 9-16
trusted CA list, sending 9-28
trusted CA path names, specifying 9-16
trusted CAs file 10-31
trusted CAs, specifying 9-4
Type of Service field
  See ToS field.

U
UC Davis agent A-55, B-10
UDP monitors A-66
UDP profile settings 8-24
UIE commands, defined 17-3
UIE function commands, listed 17-11
UIE, defined 17-1
Uniform Resource Identifier (URI) 10-40
Universal Inspection Engine, defined 17-1
universal persistence, defined 7-22
universal profile settings 7-22
unrecognized destination addresses 2-5
unused bandwidth
  borrowing 16-8
  replenishing 16-6
  saving 16-6
update interval 10-43
URI paths, redirecting 6-5
URIs
  and redirections 6-8
  including and excluding 6-13
  rewriting 6-8
URI-specified responses, managing 6-17
URL checking 10-31
user roles
  and rate classes 16-2
  and SNAT pools 14-7, 14-10
user-defined metrics, gathering A-55, B-10
username extraction search method 10-22
V
Vary headers
  enabling and disabling 6-21
  inserting 6-21
verification
  See also certificate verification, server-side.
  See certificate verification, client-side.
verification failures 10-32
verification process
  See also verification.
  See client verification process.
Via header, inserting into a SIP message 6-45
virtual bounceback 14-2
virtual server addresses, and VLANs 2-12
virtual server behavior 2-3
virtual server capabilities 2-1
virtual server mappings, defining wildcard 2-13
virtual server properties, configuring 2-14
virtual server resources
  assigning 2-14
  modifying 2-23
virtual server screen, shown 1-12
virtual server settings
  configuring 2-14
virtual server statistics, viewing 2-24
virtual server types 2-5
virtual servers
  and iRules 2-1, 17-10
  and network map 1-15
  and persistence 7-5
  and profiles 2-15, 5-12
  and RTSP protocol 6-47
  and SNATs 14-6, 14-9
  as destinations 8-11
defined 2-1
deleting 2-27
Index

disabling 2-12
for Fast HTTP profile 2-2
for Fast L4 profile 2-2
viewing 2-24
VLAN groups, creating 2-12
VMware VDI application 1-19

W
WAP monitor A-68
Warning Logging setting 10-6
web servers, and cookie generation 7-10
WebAccelerator module 8-14
Welcome screen 1-26
when keyword 17-10
wildcard servers
  assigning to VLANs 2-7
  creating 2-12
wildcard virtual servers, defined 2-6
Windows 2000 Server agent A-55, B-10
Wireless Application Protocol monitor
  See WAP monitor.
WMI monitors A-68
WMI, configuring for dynamic ratio load balancing B-3

X
X-Forwarded-For header 6-10