



3-DNS Reference Guide

version 4.2

Product Version

This manual applies to version 4.2 of the 3-DNS®.

Legal Notices

Copyright

Copyright 1998-2002, F5 Networks, Inc. All rights reserved.

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

Trademarks

F5 and the F5 logo, F5 Networks, BIG-IP, 3-DNS, GLOBAL-SITE, SEE-IT, and EDGE-FX are registered trademarks of F5 Networks, Inc. FireGuard, iControl, Internet Control Architecture, and IP Application Switch are trademarks of F5 Networks, Inc. In Japan, the F5 logo is trademark number 4386949, BIG-IP is trademark number 4435184, 3-DNS is trademark number 4435185, and SEE-IT is trademark number 4394516. All other product and company names are registered trademarks or trademarks of their respective holders.

Export Regulation Notice

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

Export Warning

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

FCC Compliance

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference.

Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

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

Canadian Regulatory Compliance

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

Standards Compliance

The product conforms to ANSI/UL Std 1950 and Certified to CAN/CSA Std. C22.2 No. 950.

Acknowledgments

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 the NetBSD Foundation, Inc. and its contributors.

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

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

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

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

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

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

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

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

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

This product includes software developed by Christos Zoulas.

This product includes software developed by Charles Hannum.

This product includes software developed by Charles Hannum, by the University of Vermont and Stage Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.

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

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD," "FreeBSD," "Mach" (by CMU).

In the following statement, "This software" refers to the parallel port driver: This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.

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 developed by Darren Reed. (© 1993-1998 by Darren Reed).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/1/gpl.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 Eric Young.

Rsync was written by Andrew Tridgell and Paul Mackerras, and is available under the Gnu Public License.

This product includes Malloc library software developed by Mark Moraes. (© 1988, 1989, 1993, University of Toronto).

This product includes open SSL software developed by Eric Young (eay@cryptsoft.com), (© 1995-1998).

This product includes open SSH software developed by Tatu Ylonen (ylo@cs.hut.fi), Espoo, Finland (© 1995).

This product includes open SSH software developed by Niels Provos (© 1999).

This product includes SSH software developed by Mindbright Technology AB, Stockholm, Sweden, www.mindbright.se, info@mindbright.se (© 1998-1999).

This product includes free SSL software developed by Object Oriented Concepts, Inc., St. John's, NF, Canada (© 2000).

This product includes software developed by Object Oriented Concepts, Inc., Billerica, MA, USA (© 2000).



Table of Contents

I		
Introduction		
	Getting started	I-1
	Using the Administrator Kit	I-1
	Stylistic conventions	I-2
	Finding help and technical support resources	I-3
2		
3-DNS Maintenance Menu		
	Working with the 3-DNS Maintenance menu	2-1
	Configuring zone files and wide IPs	2-2
	Edit BIND Configuration	2-2
	Edit 3-DNS Configuration	2-2
	Viewing statistics	2-2
	Working with the big3d agent	2-4
	Check big3d versions	2-4
	Edit big3d matrix	2-4
	Install and Start big3d	2-5
	Restart big3d	2-5
	Managing synchronized files	2-5
	Working with security issues	2-5
	Configure SSH communication with remote devices	2-5
	Generate and Copy iQuery Encryption key	2-5
	Configuring the 3-DNS Configuration utility	2-6
	Reconfigure 3-DNS Configuration Utility	2-6
	Restart 3-DNS Configuration Utility	2-6
	Change/Add Users for 3-DNS Configuration Utility	2-6
	Working with syncd	2-6
	Stop syncd	2-7
	Restart syncd	2-7
	Configuring NTP	2-7
	Configuring NameSurfer	2-7
3		
Access Control Lists		
	Working with access control lists	3-1
4		
The big3d Agent		
	Working with the big3d agent	4-1
	Setting up data collection with the big3d agent	4-1
	Collecting path data and server performance metrics	4-1
	Installing the big3d agent	4-2
	Understanding factories run by big3d agents	4-3
	Understanding the data collection and broadcasting sequence	4-5
	Tracking LDNS probe states	4-5
	Evaluating big3d agent configuration trade-offs	4-5
	Setting up communication between 3-DNS systems and other servers	4-6
	Setting up iQuery communications for the big3d agent	4-7
	Allowing iQuery communications to pass through firewalls	4-9

5

DNS Resource Records

Understanding DNS resource records	5-1
Types of resource records	5-2
A (Address)	5-2
CNAME (Canonical Name)	5-2
MX (Mail Exchange)	5-3
NS (Name Server)	5-3
PTR (Pointer)	5-3
SOA (Start of Authority)	5-4
Additional resource record types	5-5

6

Extended Content Verification (ECV)

Working with the ECV Service Monitor	6-1
Defining ECV service monitors	6-1
Using the search string option	6-2

7

Internet Weather Map

Overview of the Internet Weather Map	7-3
Working with the Average Round Trip Time table	7-4
Working with the Average Completion Rate table	7-4
Working with the Average Router Hops table	7-5
Interpreting the Internet Weather Map data	7-5

8

Load Balancing

Working with load balancing modes	8-1
Understanding load balancing on the 3-DNS	8-1
Using static load balancing modes	8-2
Using dynamic load balancing modes	8-5
Configuring load balancing	8-9
Understanding wide IPs	8-10
Understanding pools	8-10
Defining a wide IP	8-11
Using wildcard characters in wide IP names	8-12
An example of the wideip statement	8-13
Using the LDNS round robin wide IP attribute	8-14
Using the last resort pool designation	8-14
Changing global variables that affect load balancing	8-15
Setting global alternate and fallback methods	8-16
Understanding TTL and timer values	8-17
Setting up load balancing for services that require multiple ports	8-20
An example configuration using a port list	8-21
Troubleshooting manual configuration problems	8-22

9

Network Map

Introducing the Network Map	9-1
Working with the Network Map	9-2
Viewing the Network Map	9-2
Using the Network Map to review and modify the network configuration	9-2
Using the information table on the Network Map	9-2

10

Production Rules

Controlling network traffic patterns with production rules	10-1
Setting up production rules in the Configuration utility	10-1
Viewing, adding, and deleting production rules	10-2
Choosing the rule type	10-2
Defining time-based triggers	10-2
Defining event-based triggers	10-4
Choosing the action taken	10-4
Working with the production rules scripting language	10-5
Inserting production rules in the wideip.conf file	10-5
Executing and managing production rules from the command line	10-5
Working with the if statement	10-6
Working with the when statement	10-7
Working with the every statement	10-8
Defining production rule actions	10-9
Production rule examples	10-10

11

Scripts

Working with scripts	11-1
3dns_add script	11-1
3dns_admin_start script	11-1
3dns_dump script	11-1
3dns_web_config script	11-1
3dns_web_passwd script	11-1
3dnsmaint script	11-2
3dprint script	11-2
3ndc script	11-3
big3d_install script	11-3
big3d_restart script	11-4
big3d_version script	11-4
config_ssh script	11-4
edit_lock script	11-4
edit_wideip script	11-4
install_key script	11-5
syncd_checkpoint script	11-5
syncd_rollback script	11-6
syncd_start script	11-7
syncd_stop script	11-7

12

SNMP

Working with SNMP on the 3-DNS	12-1
Configuring SNMP on the 3-DNS	12-1
Downloading the MIBs	12-2
Understanding configuration file requirements	12-2
Configuring options for the checktrap.pl script	12-6
Configuring the 3-DNS SNMP agent using the Configuration utility	12-6
Configuring host SNMP settings on the 3-DNS	12-7
Configuring the SNMP agent on host servers	12-10

13

Topology

Working with Topology load balancing	13-1
Setting up topology records	13-1
Using the Topology load balancing mode in a wide IP	13-3
Using the Topology load balancing mode in a pool	13-4
Understanding user-defined regions	13-5
Working with the topology statement in the wideip.conf file	13-6

A

3-DNS Configuration File

Overview of the 3-DNS configuration file	A-1
Using include files	A-2
Syntax for include files	A-2
Working with statements	A-3
Syntax rules	A-4
The globals statement	A-5
The server statement	A-18
The datacenter statement	A-29
The sync_group statement	A-31
The wide IP statement	A-32
The topology statement	A-39
Access control lists	A-40
Working with comments	A-42
Syntax	A-42
Definition and usage	A-42
Understanding current values	A-43
Server definition current values	A-43
Virtual server definition current values	A-44
Local DNS server paths current values	A-45
Wide IP definition current values	A-46

B**3dpipe Command Reference**

3dpipe commands	B-1
datacenter (or dc)	B-3
-help (or -h)	B-4
<server type>	B-5
stats	B-6
syncgroup (or sg)	B-7
-version (or -v)	B-8
virtual (or vs)	B-9
wideip (or wip)	B-10

C**bigpipe Command Reference**

bigpipe commands	C-1
-?	C-3
config	C-4
Saving configuration files to an archive	C-4
Installing an archived configuration file	C-4
failover	C-5
global	C-6
-h and -help	C-9
interface	C-10
Setting the media type	C-10
Setting the duplex mode	C-10
load	C-11
merge	C-12
monitor	C-13
Showing, disabling, and deleting monitors	C-13
reset	C-14
save	C-15
self	C-16
trunk	C-17
Creating a trunk	C-17
unit	C-18
verify	C-19
version	C-20
vlan	C-21
Creating and assigning a VLAN	C-22
Tagged VLANs	C-22
Enabling and disabling port lockdown	C-23
Setting the fail-over timeout and arming the fail-safe	C-23
Setting the MAC masquerade address	C-23
vlangroup	C-25

Glossary**Index**

Table of Contents



I

Introduction

- Getting started
- Using the Administrator Kit
- Finding help and technical support resources

Getting started

The *3-DNS Reference Guide* includes information about the features of the 3-DNS® system. It also contains information about system configuration files and variables, command line syntax, scripts and utilities, and other 3-DNS objects. Use the *3-DNS Reference Guide* for help in configuring a specific feature of the 3-DNS. For load balancing and networking solutions, see the *3-DNS Administrator Guide*.

Using the Administrator Kit

The 3-DNS Administrator Kit provides simple steps for quick, basic configuration, and also provides detailed information about more advanced features and tools, such as the **3dnsmaint** command line utility. The information is organized into the guides described as follows.

- ◆ **Configuration Worksheet**
Use the Configuration Worksheet to gather the IP addresses, default routes, administrative account, and server information you need to configure the 3-DNS. The Setup utility prompts you for this information when you configure the 3-DNS for the first time.
- ◆ **Hardware poster**
The hardware poster is a graphical representation of the physical components of the 3-DNS.
- ◆ **3-DNS Administrator Guide**
The *3-DNS Administrator Guide* provides examples of common wide-area load balancing solutions supported by the 3-DNS. For example, in the Administrator Guide, you can find everything from a basic DNS request load balancing solution to a more advanced content acceleration load balancing solution. The Administrator Guide also covers general network administration issues, such as installing the hardware and setting up the networking configuration.
- ◆ **3-DNS Reference Guide**
The *3-DNS Reference Guide* provides basic descriptions of individual 3-DNS objects, such as wide IPs, pools, virtual servers, load balancing modes, the **big3d** agent, resource records, and production rules. It also provides syntax information for **3dnsmaint** commands, configuration utilities, the **wideip.conf** file, and system utilities.

◆ **Note**

*If you are configuring the 3-DNS module on the BIG-IP, use the **BIG-IP Reference Guide** and hardware poster to set up and configure the hardware.*

Stylistic conventions

To help you easily identify and understand certain types of information, this documentation uses the stylistic conventions described below.

◆ WARNING

All examples in this documentation use only non-routable IP addresses. When you set up the solutions we describe, you must use IP addresses suitable to your own network in place of our sample IP addresses.

Identifying new terms

When we first define a new term, the term is shown in bold italic text. For example, a ***virtual server*** is the combination of an IP address and port that maps to a set of back-end servers.

Identifying references to objects, names, and commands

We make a variety of items bold 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, the **nslookup** command requires that you include at least one **<ip_address>** variable.

Identifying references

We use italic text to denote a reference to another document or another section in the current document. In references where we provide the name of a book as well as a specific chapter or section in the book, we show the book name in bold, italic text, and the chapter/section name in italic text to help quickly differentiate the two. For example, you can find information about planning the 3-DNS configuration in the ***3-DNS Administrator Guide***, Chapter 2, *Planning the 3-DNS Configuration*.

Identifying command syntax

We show actual, 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 current status of the 3-DNS daemons:

```
3ndc status
```


Table 1.1 explains additional special conventions used in command line syntax.

Item in text	Description
\	Continue to the next line without typing a line break.
< >	You enter text for the enclosed item. For example, if the command has <your name>, type in your name.
	Separates parts of a command.
[]	Syntax inside the brackets is optional.
...	Indicates that you can type a series of items.

Table 1.1 Command line conventions used in this manual

Finding help and technical support resources

You can find additional technical documentation about the 3-DNS in the following locations:

- ◆ **Release notes**

The release note for the current version of the 3-DNS is available from the home page of the Configuration utility. The release note contains the latest information for the current version, including a list of new features and enhancements, a list of fixes, and a list of known issues.

- ◆ **Online help for 3-DNS features**

You can find help online in three different locations:

- The Configuration utility home page has PDF versions of the guides included in the Administrator Kit. Software upgrades for the 3-DNS replace the guides with updated versions as appropriate.
- The Configuration utility has online help for each screen. Just click the **Help** button in the toolbar.
- Individual commands have online help, including command syntax and examples, in standard UNIX man page format. Simply type **man** followed by the command (for example **man 3dpipe**), and the 3-DNS displays the syntax and usage associated with the command.

- ◆ **Third-party documentation for software add-ons**

The Configuration utility contains online documentation for the third-party software included with the 3-DNS, including NameSurfer and GateD.



2

3-DNS Maintenance Menu

- Working with the 3-DNS Maintenance menu
- Configuring zone files and wide IPs
- Viewing statistics
- Working with the big3d agent
- Managing synchronized files
- Working with security issues
- Configuring the 3-DNS Configuration utility
- Working with syncd
- Configuring NTP
- Configuring NameSurfer

Working with the 3-DNS Maintenance menu

The 3-DNS Maintenance menu is a utility that you can use to configure and monitor the 3-DNS from the command line. You can perform the following tasks:

- Edit the **wideip.conf** configuration file
- Edit the BIND configuration files
- View statistics
- Work with the **big3d** agent
- Manage synchronized files
- Work with security issues
- Configure the 3-DNS web server
- Work with **syncd**
- Configure NTP
- Configure NameSurfer

◆ WARNING

*If you use the browser-based NameSurfer application, you cannot use the **Edit BIND Configuration** command on the 3-DNS Maintenance menu to configure your DNS zone files.*

Figure 2.1 shows the main screen of the 3-DNS Maintenance menu.

```
3 D N S(®) Maintenance Menu

Configure SSH communication with remote devices
Generate and Copy iQuery Encryption Key
Check remote versions of big3d
Edit big3d matrix
Install and Start big3d
Edit BIND Configuration
Edit 3-DNS Configuration
Backup the 3-DNS
Restore a 3-DNS from a backup
Synchronize Metrics Data
Restart big3d
Reconfigure 3-DNS Configuration Utility
Restart 3-DNS Configuration Utility
Change/Add Users for 3-DNS Configuration Utility
Dump 3dnsd Statistics
Stop syncd
Restart syncd
Configure connection to NTP time server
Configure NameSurfer(TM)
Enter 'q' to Quit
```

Figure 2.1 The 3-DNS Maintenance menu main screen

To use the 3-DNS Maintenance menu from the command line

1. On the command line, type the following command to open the menu:
`3dnsmaint`
2. From the menu, choose the command to you wish to run, and press the Enter key.

Each command is described in the following sections.

Configuring zone files and wide IPs

We recommend that you use NameSurfer to configure BIND zone files, and that you use the Configuration utility to configure wide IPs. However, if you choose to edit the BIND zone files and the 3-DNS configuration files from the command line, use the following commands.

Edit BIND Configuration

The **Edit BIND Configuration** command opens the **named.conf** file for editing.

◆ WARNING

Use this command only if you are performing all configuration tasks from the command line. It is important that you do not use this command if you are using NameSurfer.

Edit 3-DNS Configuration

The **Edit 3-DNS Configuration** command runs the **edit_wideip** script, which performs the following tasks:

- Opens the **wideip.conf** file for editing
- Copies the **wideip.conf** file to all other 3-DNS systems in the local system's sync group
- Restarts **3dnsd**

Viewing statistics

From the Maintenance menu, use the **Dump 3dnsd Statistics** command to view various 3-DNS statistics. The **Dump 3dnsd Statistics** command corresponds to the **3dprint** script, which lets you view the following statistics screens at the command line:

- ◆ **3-DNS**

This object displays statistics about each 3-DNS in your network. The statistics include such things as whether the 3-DNS is enabled or disabled, the number of packets per second traveling in and out of the 3-DNS during the last sample period, the name of the sync group to which each 3-DNS belongs, and so on.
- ◆ **BIG-IP**

This object displays statistics about all BIG-IP systems known to the 3-DNS. The statistics include such things as the number of virtual servers each BIG-IP manages, the number of times the 3-DNS resolves requests to those virtual servers, and more.
- ◆ **EDGE-FX**

This object displays statistics about all EDGE-FX Caches known to the 3-DNS. The statistics include such things as the number of virtual servers each EDGE-FX Cache manages, the number of times the 3-DNS resolves requests to those virtual servers, and more.
- ◆ **Hosts**

This object displays statistics about all hosts known to the 3-DNS, such as the number of times the 3-DNS resolves requests to the host, and the number of virtual servers that the hosts manage.
- ◆ **Virtual Servers**

This object displays statistics about BIG-IP, EDGE-FX Cache, and host virtual servers; the statistics include such things as the server state, and the number of times it has received resolution requests.
- ◆ **Paths**

This object displays path statistics such as round trip time, packet completion rate, the remaining time to live (TTL) before a path's metric data needs to be refreshed, and so on.
- ◆ **Local DNS**

This object displays statistics collected for LDNS servers: the number of resolution requests received from a given server, the current protocol used to probe the server, and more.
- ◆ **Wide IPs**

This object displays statistics about each wide IP defined on the 3-DNS. The statistics include such things as load balancing information, the remaining time to live (TTL) before the wide IP's metrics data needs to be refreshed, and so on.
- ◆ **Globals**

This object displays statistics about the globals sub-statements. The statistics include such things as the current and default values for each of the globals sub-statements, and whether you have to restart **3dnsd** when you make changes to the parameters.
- ◆ **Summary**

This object displays summary statistics such as the 3-DNS version, the total number of resolved requests, and the load balancing methods used to resolve requests.

- ◆ **Data Centers**

This object displays statistics about the data centers and their servers in your network. The statistics include such things as the names of the data centers, the name or IP address of the servers in the data center, and whether the data center is enabled or disabled.

- ◆ **Sync Groups**

This object displays statistics about each sync group in your network. The statistics include such things as the name of the sync group, whether **3dnsd** is running on each 3-DNS, whether the **big3d** agent is running on each 3-DNS, the name and IP address of the 3-DNS, and whether the 3-DNS is a principal or receiver.

To view more statistics information, expand the **Statistics** item on the navigation pane in the Configuration utility.

Working with the big3d agent

You can use the following commands to work with the **big3d** agent, which collects information about paths between a data center and a specific local DNS server.

Check big3d versions

The **Check remote versions of big3d** command runs the **big3d_version** script. This script checks that the correct version of **big3d** is running on all BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems known to the 3-DNS.

Edit big3d matrix

The **Edit big3d matrix** command opens an editable file that lists version numbers, and the appropriate **big3d** agent, for all BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems known to the 3-DNS.

You do not need to edit this file unless a new version of BIG-IP, EDGE-FX Cache, or GLOBAL-SITE creates a conflict. If this happens, you need to place a new version of the **big3d** agent on all affected servers.

The **Install and Start big3d** command uses the matrix file to determine which version of the **big3d** agent to transfer to the BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems.

Install and Start big3d

The **Install and Start big3d** command runs the **big3d_install** script, which installs and starts the appropriate version of the **big3d** agent on each BIG-IP, EDGE-FX Cache, and GLOBAL-SITE in the network.

Restart big3d

The **Restart big3d** command runs the **big3d_restart** script, which stops and restarts the **big3d** agent on each BIG-IP, EDGE-FX Cache, and GLOBAL-SITE.

Managing synchronized files

You can use the following commands to copy metrics data to a new 3-DNS, to archive synchronized files, or to retrieve an archive.

Working with security issues

You can use the following commands to address security issues for your network setup.

Configure SSH communication with remote devices

The **Configure SSH communication with remote devices** command runs the **config_ssh** script, which configures secure shell access to any new 3-DNS, BIG-IP, EDGE-FX Cache, or GLOBAL-SITE that is added to a network.

For more information, see Chapter 11, *Scripts*.

Generate and Copy iQuery Encryption key

The **Generate and Copy iQuery Encryption key** command runs the **install_key** script, which then runs the **F5makekey** program. The **F5makekey** program generates a seed key for encrypting communications between the 3-DNS and BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems.

For more information, see Chapter 11, *Scripts*.

◆ **Note**

This command is not available on the non-crypto version of the 3-DNS.

Configuring the 3-DNS Configuration utility

You can use the following commands to configure the 3-DNS Configuration utility, which is hosted by the 3-DNS web server.

Reconfigure 3-DNS Configuration Utility

The **Reconfigure 3-DNS Configuration Utility** command runs the **config_httpd** script, which lets you make configuration changes to the 3-DNS web server.

Restart 3-DNS Configuration Utility

The **Restart 3-DNS Configuration Utility** command runs the **3dns_admin_start** script, which restarts the 3-DNS web server.

Change/Add Users for 3-DNS Configuration Utility

The **Change/Add Users for 3-DNS Configuration Utility** command runs the **3dns_web_passwd** script, which lets you provide one of three levels of access to the 3-DNS Configuration utility for selected users only, and assign passwords for those users. The three levels of user access are:

- ◆ **Read-only**
Users with this level of access can only view the configuration and statistics information in the Configuration utility.
- ◆ **Partial read/write**
Users with this level of access can view configuration and statistics information in the Configuration utility. They can also enable and disable objects in the configuration.
- ◆ **Full read/write**
Users with this level of access have full administrative access to all components of the Configuration utility.

You can also add, remove, and modify users and their administrative access levels using the Configuration utility. For more information, see the *3-DNS Administrator Guide*, Chapter 12, *Administration and Monitoring*.

Working with syncd

You can use the following commands to work with **syncd**, the synchronization daemon that runs on all 3-DNS systems. The function of **syncd** is to update and synchronize all 3-DNS configuration files.

Stop syncd

The **Stop syncd** command runs the **syncd_stop** script, which stops the **syncd** daemon, if it is running.

Restart syncd

The **Restart syncd** command runs the **syncd_start** script, which restarts the **syncd** daemon if it is already running, or starts it if it is not.

Configuring NTP

The 3-DNS systems in a network must have their time synchronized to within a few seconds of each other. If you do not synchronize the systems, it is done by default through iQuery messages exchanged between 3-DNS systems. However, the following command allows much more precise time synchronization between the 3-DNS systems.

The **Configure Connection to NTP Time Server** command allows the 3-DNS to synchronize its time to a public NTP (Network Time Protocol) server on the Internet. To simplify the task of choosing the best time server, this command has a list of regional time servers built into it. A 3-DNS is not required to have NTP configured; depending on the network configuration, it may not be possible to configure NTP (for example, if the 3-DNS is behind a firewall and the firewall does not pass NTP packets).

Configuring NameSurfer

The **Configure NameSurfer** command makes NameSurfer the primary name server on the 3-DNS. NameSurfer then handles zone file management, and processes all changes and updates to the zone files. Note that configuring NameSurfer as the primary name server for your domains is an optional setting. You access the NameSurfer application in the Configuration utility by clicking **NameSurfer** in the navigation pane. Note that you only use NameSurfer if you configure the 3-DNS in node mode.

◆ **Note**

If you do not set NameSurfer to be the primary name server for your wide IP zones, (when you run the 3-DNS in node mode only) you must maintain all of your zone file information manually.



3

Access Control Lists

Working with access control lists

With access control lists (ACLs), you can block probing for members of the ACL when you use dynamic RTT probing on your 3-DNS. Table 3.1 lists the ACL types and describes their functions.

ACL Type	Description
Prober	Prober ACLs limit round-trip time probes.
Hops	Hops ACLs limit traceroute probes.

Table 3.1 Access control list types and descriptions

To define ACLs using the Configuration utility

1. In the navigation pane, click **System**.
The System - General screen opens.
2. On the toolbar, click **ACL**.
The ACL Configuration screen opens.
3. Add the settings for the ACLs you want to create, and click **Update**.
For more information on this screen, click **Help** on the toolbar.

To define ACLs from the command line

1. If one does not already exist, create a file called **region.ACL** in the **/var/3dns/include** directory. You must add the **include** file at the beginning of the **wideip.conf** file.
2. Add the file to **/etc/wideip.conf** by typing, at the command line:

```
include "region.ACL"
```

◆ Tip

*When you create ACLs by editing the **wideip.conf** file from the command line, we strongly recommend that you put the ACLs in a separate **include** file.*

The ACLs you can create are **probe_acl**, and **hops_acl**. Figure 3.1 is an example of the syntax for a **region.ACL** file with definitions for the two ACL types.

```
actions {
  NO_RELAY
  delete rdb ACL region "probe_acl"
  delete rdb ACL region "hops_acl"
}
region_db ACL {
  region {
    name "probe_acl"
    region "probe_acl"
    192.168.4.0/24
  }
  region {
    name "hops_acl"
    192.168.2.0/16
  }
}
```

Figure 3.1 Sample region.ACL file



4

The big3d Agent

- Working with the big3d agent
- Installing the big3d agent
- Understanding factories run by big3d agents
- Understanding the data collection and broadcasting sequence
- Setting up communication between 3-DNS systems and other servers

Working with the big3d agent

The **big3d** agent collects performance information on behalf of the 3-DNS. The **big3d** agent runs on 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems; the default setting is to run a **big3d** agent on all of these systems in the network, but you can turn off the **big3d** agent on any system at any time.

Setting up data collection with the big3d agent

Setting up the **big3d** agents involves the following tasks:

- ◆ **Installing big3d agents on BIG-IP, EDGE-FX Cache, and GLOBAL-SITE**

Each new version of the 3-DNS software includes the latest version of the **big3d** agent. You need to distribute that copy of the **big3d** agent to each BIG-IP, EDGE-FX Cache, and GLOBAL-SITE in the network. See the release notes provided with the 3-DNS for information about which BIG-IP, EDGE-FX Cache, and GLOBAL-SITE versions the current **big3d** agent supports. For details on installing the **big3d** agent, see *Installing the big3d agent*, on page 4-2.

- ◆ **Specifying which factories a specific big3d agent manages**

When you define 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems in the configuration, you can change the default **big3d** agent settings by changing the factories settings on a specific system. You can change the number of factories the **big3d** agent runs, and turn specific factories on and off. For more information on factories, see *Understanding factories run by big3d agents*, on page 4-3.

- ◆ **Setting up communications between big3d agents and other systems**

Before the **big3d** agents can communicate with the 3-DNS systems in the network, you need to configure the appropriate ports and tools to allow communication between the devices running the **big3d** agent and 3-DNS systems in the network. These planning issues are discussed in *Setting up communication between 3-DNS systems and other servers*, on page 4-6.

Collecting path data and server performance metrics

A **big3d** agent collects the following types of performance information used for load balancing. This information is broadcast to all 3-DNS systems in your network.

- ◆ **Virtual server availability**

The **big3d** agent queries virtual servers to verify whether they are up and available to receive connections. For name resolution, the 3-DNS uses only those virtual servers that are **up**.

- ◆ **Network path round trip time**

The **big3d** agent calculates the round trip time for the network path between the data center and the client's LDNS server that is making the

resolution request. The round trip time is used to determine the best virtual server to answer the request when you use the Round Trip Times or the Quality of Service load balancing modes.

◆ **Network path packet loss**

The **big3d** agent calculates the packet completion percentage for the network path between the data center and the client's LDNS server that is making the resolution request. Packet completion is used to determine the best virtual server to answer the request when you use the Completion Rate or the Quality of Service load balancing modes.

◆ **Router hops along the network path**

The **big3d** agent calculates the number of intermediate systems transitions (router hops) between the data center and the client's LDNS server. Hops are used to determine the best virtual server to answer the request when you use the Hops or the Quality of Service load balancing modes.

◆ **Server performance**

The **big3d** agent calculates server metrics, such as the packet rate for BIG-IP systems or SNMP-enabled hosts. Packet rate is used to determine the best virtual server to answer the request when you use the Packet Rate or the Quality of Service load balancing modes.

◆ **Virtual server performance**

The **big3d** agent calculates the number of connections to virtual servers defined on BIG-IP systems or SNMP-enabled hosts. The number of virtual server connections is used to determine the best virtual server when using the Least Connections load balancing mode.

Installing the big3d agent

You can easily install the **big3d** agent on the BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems in your network by using the 3-DNS Maintenance menu.

To install the big3d agent from the command line

1. Log on to the 3-DNS using either a remote shell, a serial terminal, or a keyboard and monitor attached directly to the system.
2. At the command prompt, type **3dnsmaint**.
The 3-DNS Maintenance menu opens.
3. Choose the **Install and Start big3d** command from the menu, and press Enter.

Understanding factories run by big3d agents

To gather performance information, the **big3d** agent uses different types of factories. A *factory* is a process that collects different types of data. The **big3d** agent currently supports the following factory types:

- ◆ **Prober factory**

A prober factory collects several types of information using ICMP, TCP, UDP, DNS_DOT, or DNS_REV protocols. This factory queries host virtual servers and local DNS servers. Host virtual servers are checked to determine their **up** or **down** state. For local DNS servers, the prober factory uses the response time to calculate the round trip time and packet loss between the LDNS and the data center.

- ◆ **Hops factory**

A hops factory uses the traceroute method to calculate the number of intermediate systems transitions (or router hops) along the network path between a specific data center and a client LDNS.

- ◆ **SNMP factory**

An SNMP factory queries the SNMP agents that run on host servers to collect performance metrics for the host.

- ◆ **ECV factory**

When you have set up extended content verification (ECV) service monitors for wide IPs, an ECV factory performs a more extensive availability check than the prober factories. (For more information on ECV service monitors, see Chapter 6, *Extended Content Verification (ECV)*).

The standard configuration specifies that each 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE in the network run a **big3d** agent using five prober factories, one SNMP factory, no hops factories, and five ECV factories. You can change the number of factories that the **big3d** agent runs either by using the Configuration utility, or by editing the server definition in the **wideip.conf** file.

To edit the factory settings for a 3-DNS using the Configuration utility

1. In the navigation pane, click **Servers**, and then click **3-DNS**. The 3-DNS List screen opens.
2. In the list, click the name of the 3-DNS that you want to modify. The Modify 3-DNS screen opens.
3. Make the changes to the factory settings that you want to make, and click **Update**. For more information on the settings on this screen, click **Help** on the toolbar.

To edit the factory settings for a BIG-IP using the Configuration utility

1. In the navigation pane, click **Servers**, and then click **BIG-IP**.
The BIG-IP List screen opens.
2. In the list, click the name of the BIG-IP that you want to modify.
The Modify BIG-IP screen opens.
3. Make the changes to the factory settings that you want to make, and click **Update**. For more information on the settings on this screen, click **Help** on the toolbar.

To edit the factory settings for an EDGE-FX Cache using the Configuration utility

1. In the navigation pane, click **Servers**, and then click **EDGE-FX Cache**.
The EDGE-FX Cache List screen opens.
2. In the list, click the name of the EDGE-FX Cache that you want to modify.
The Modify EDGE-FX Cache screen opens.
3. Make the changes to the factory settings that you want to make, and click **Update**. For more information on the settings on this screen, click **Help** on the toolbar.

To edit the factory settings for a GLOBAL-SITE using the Configuration utility

1. In the navigation pane, click **Servers**, and then click **GLOBAL-SITE**.
The GLOBAL-SITE List screen opens.
2. In the list, click the name of the GLOBAL-SITE that you want to modify.
The Modify GLOBAL-SITE screen opens.
3. Make the changes to the factory settings that you want to make, and click **Update**. For more information on the settings on this screen, click **Help** on the toolbar.

To edit the factory settings from the command line

1. From the command line, type **3dnsmaint**.
The 3-DNS Maintenance menu opens.
2. Select **Edit 3-DNS Configuration**, and press Enter.
3. Find the server definition that you want to modify and make your changes. For an example, see any of the server definitions in Appendix A, *3-DNS Configuration File*.

Understanding the data collection and broadcasting sequence

The **big3d** agents collect and broadcast information on demand. The principal 3-DNS in a sync group issues a data collection request to all **big3d** agents running in the network. In turn, the **big3d** agents collect the requested data using factories, and then broadcast that data to all 3-DNS systems running in the network, including the principal 3-DNS that issued the request.

Tracking LDNS probe states

The 3-DNS tracks the state of path data collection for each LDNS that has ever requested a name resolution from the system. Table 4.1 shows the states that can be assigned to an LDNS. Note that you can view the state of LDNS servers in the Local DNS Statistics screen in the Configuration utility.

State	Description
Needs Probe	The big3d agent has never collected data for the LDNS, or the data has expired.
Idle	The big3d agent successfully collected data for the LDNS, and is waiting for the next collection request.
In Probe	The big3d agent is currently collecting data for the LDNS.

Table 4.1 Probe and discovery states for individual client LDNS servers

Evaluating big3d agent configuration trade-offs

You must run a **big3d** agent on each BIG-IP, 3-DNS, EDGE-FX Cache, and GLOBAL-SITE if you are using dynamic load balancing modes (those that rely on path data) on the 3-DNS. You must have a **big3d** agent running on at least one system in each data center to gather the necessary path metrics.

The load on the **big3d** agents depends on two factors: the timer settings that you assign to the different types of data the **big3d** agents collect, and the number of factories that each **big3d** agent runs. The shorter the timers, the more frequently the **big3d** agent needs to refresh the data. While short timers guarantee that you always have valid data readily available for load balancing, they also increase the frequency of data collection. The more factories a **big3d** agent runs, the more metrics it can refresh at one time, and the more quickly it can refresh data for the 3-DNS.

Another factor that can affect data collection is the number of client LDNS servers that make name resolution requests. The more LDNS servers that make resolution requests, the more path data that the **big3d** agents have to collect. While round trip time for a given path may vary constantly due to current network load, the number of hops along a network path between a

data center and a specific LDNS does not often change. Consequently, you may want to set short timer settings for round trip time data so that it refreshes more often, but set high timer settings for hops data because it does not need to be refreshed often.

Setting up communication between 3-DNS systems and other servers

In order to copy **big3d** agents from a 3-DNS to BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems, the 3-DNS must be able to communicate with those systems. If you use exclusively crypto systems, or exclusively non-crypto systems, the communication tools you configure when you run the Setup utility are all you need. Crypto systems all use **ssh** and **scp**, and non-crypto systems all use **rsh** and **rcp**.

However, if your network is a mixed environment, where some systems are crypto and other systems are non-crypto, you need to enable the **rsh** and **rcp** tools on the crypto systems so that they can communicate with the non-crypto systems. These tools are pre-installed on all crypto systems, however, you must explicitly enable them.

To enable RSH on a crypto system from the command line

1. Type **config**, and press Enter.
The Setup utility opens.
2. From the menu, select **(R) Configure RSH**, and press Enter.
3. Follow the onscreen instructions to enable the **rsh** and **rcp** tools.

◆ Note

*You can disable **rsh** and **rcp** access at any time by following these same steps.*

Table 4.2 shows the ports and protocols that 3-DNS uses to communicate with crypto and non-crypto BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems. Note that GLOBAL-SITE is only available as a crypto system.

From	To	Protocol	From Port	To Port	Purpose
Crypto 3-DNS	Crypto BIG-IP, Crypto EDGE-FX Cache, GLOBAL-SITE	TCP	<1024	22	SSH/SCP
Non-crypto 3-DNS	Non-crypto BIG-IP, Non-crypto EDGE-FX Cache	TCP	<1024	514	RSH/RCP

Table 4.2 Communications between 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE

From	To	Protocol	From Port	To Port	Purpose
Crypto 3-DNS	Non-crypto BIG-IP, Non-crypto EDGE-FX Cache	TCP	<1024	514	RSH/RCP
Non-crypto BIG-IP, Non-crypto EDGE-FX Cache	Crypto 3-DNS	N/A	N/A	N/A	N/A

Table 4.2 Communications between 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE

Note that if you run **big3d** agents in a mixed crypto/non-crypto environment, the crypto systems automatically turn off Blowfish encryption when communicating with non-crypto systems. When communicating with crypto systems, however, crypto 3-DNS systems use Blowfish encryption after the iQuery encryption key has been copied to all crypto 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems.

To create and distribute the iQuery encryption key from the command line

1. From the command line, type **3dnsmaint**.
The 3-DNS Maintenance menu opens.
2. Select **Generate and Copy iQuery Encryption Key**, and press Enter.
3. Follow the onscreen instructions to generate and copy the iQuery encryption key to the crypto systems in your network.

Setting up iQuery communications for the big3d agent

The iQuery protocol uses one of two ports to communicate between the **big3d** agents and 3-DNS systems. The ports used by iQuery traffic change, depending on whether the traffic is inbound from the **big3d** agent or outbound from the 3-DNS.

Table 4.3 shows the protocols, ports, and iQuery settings for both inbound and outbound iQuery communications between 3-DNS systems and **big3d** agents distributed in your network.

From	To	Protocol	From Port	To Port	Multiplex?	Use Alternate Port?
3-DNS	big3d agent	UDP	4353	4353	Yes	Yes
3-DNS	big3d agent	UDP	4354	4353	No	Yes
3-DNS	big3d agent	UDP	245	245	Yes	No
3-DNS	big3d agent	UDP	4354	245	No	No

Table 4.3 Communication protocols and ports between 3-DNS systems and **big3d** agents

From	To	Protocol	From Port	To Port	Multiplex?	Use Alternate Port?
3-DNS	big3d agent	TCP	4354	4353	Yes	Yes or No
3-DNS	big3d agent	TCP	>1023	4353	No	Yes or No
big3d agent	3-DNS	UDP	4353	4353	Yes	Yes
big3d agent	3-DNS	UDP	4353	4354	No	Yes
big3d agent	3-DNS	UDP	245	245	Yes	No
big3d agent	3-DNS	UDP	245	4354	No	No
big3d agent	3-DNS	TCP	4353	4354	Yes	Yes or No
big3d agent	3-DNS	TCP	4353	>1023	No	Yes or No

Table 4.3 Communication protocols and ports between 3-DNS systems and **big3d** agents

You can configure the multiplex and alternate port globals using the Configuration utility.

To configure the multiplex and alternate port settings using the Configuration utility

1. In the navigation pane, click System.
The System - General screen opens.
2. Check the **iQuery Settings, Use Alternate Port (port 4353)** box specify that iQuery traffic use port **4353** (the preferred, registered port). Clear the check box if you want iQuery traffic to use the old port, **245**.
3. Check the **iQuery Settings, Multiplex** box if you want UDP-based iQuery traffic to be sent and received on the same port (**245** or **4353**), and you want traffic from the **big3d** agent to use port **4354**.
4. For more information, click **Help** on the toolbar.

Table 4.4 shows the protocols and corresponding ports used for iQuery communications between **big3d** agents and SNMP agents that run on host servers.

From	To	Protocol	From Port	To Port	Purpose
big3d agent	host SNMP agent	UDP	>1023	161	Ephemeral ports used to make SNMP queries for host statistics
host SNMP agent	big3d agent	UDP	161	>1024	Ephemeral ports used to receive host statistics using SNMP

Table 4.4 Communication protocols and ports between **big3d** agents and SNMP agents

If you run a **big3d** agent on a 3-DNS, or a BIG-IP, and you set the SNMP prober factory count to **1** or higher, the **big3d** agent automatically opens the appropriate UDP ports to allow for SNMP communications. If you do not want to open the UDP ports for this purpose, you need to set the SNMP factory count to **0**.

Allowing iQuery communications to pass through firewalls

The payload information of an iQuery packet contains information that potentially requires translation when there is a firewall in the path between the **big3d** agent and the 3-DNS. The firewall translates only the packet headers, not the payloads.

The virtual server translation option resolves this issue. With virtual server translation configured, the iQuery packet stores the original IP address in the packet payload itself. When the packet passes through a firewall, the firewall translates the IP address in the packet header normally, but the IP address within the packet payload is preserved. The 3-DNS reads the IP address out of the packet payload, rather than out of the packet header.

In the example configuration shown in Figure 4.1, a firewall separates the path between a BIG-IP running a **big3d** agent and the 3-DNS. The packet addresses are translated at the firewall. However, addresses within the iQuery payload are not translated, and they arrive at the BIG-IP in their original states.

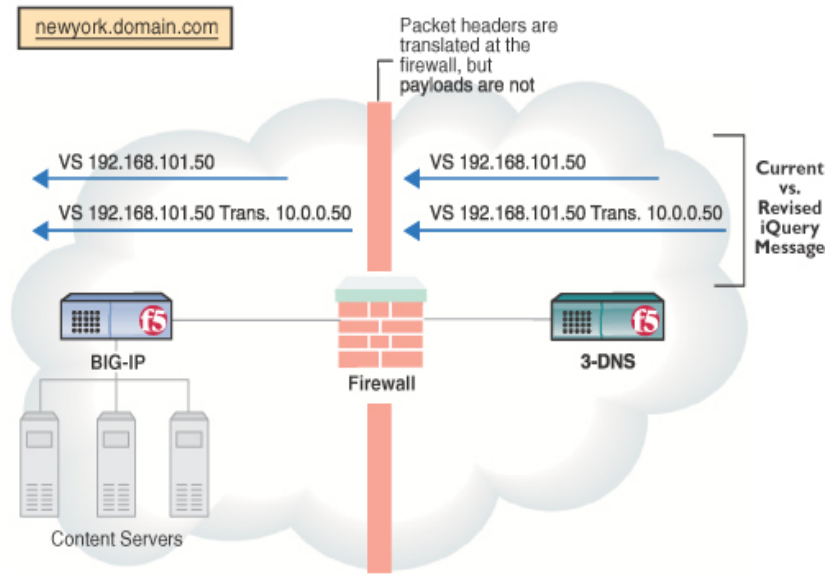


Figure 4.1 Translating packet address through the firewall

Communications for remote administration of the 3-DNS

The Configuration utility is a browser-based utility through which you can administer the 3-DNS from a remote workstation. Table 4.5 shows the ports that are used for remote administrative connections to the 3-DNS web server, which hosts the Configuration utility.

From	To	Protocol	Port	Purpose
Configuration utility on a remote workstation	Crypto 3-DNS	https over TCP	443	Connection to secure web server
Configuration utility on a remote workstation	Non-crypto 3-DNS	http over TCP	80	Connection to standard web server

Table 4.5 Communications between 3-DNS systems and remote workstations

Communications between 3-DNS, big3d agents, and local DNS servers

Table 4.6 shows the ports on which the 3-DNS receives and responds to DNS resolution requests issued by local DNS servers.

From	To	Protocol	From Port	To Port	Purpose
LDNS	3-DNS	UDP	53 or >1024	53	DNS resolution requests
3-DNS	LDNS	UDP	53	53 or >1024	DNS resolution responses

Table 4.6 DNS communications on the 3-DNS

Table 4.7 shows the protocols and ports that the **big3d** agent uses when collecting path data for local DNS servers.

From	To	Protocol	From Port	To Port	Purpose
big3d	LDNS	ICMP	N/A	N/A	Probe using ICMP pings
big3d	LDNS	TCP	>1023	53	Probe using TCP (Cisco routers: allow establish)
LDNS	big3d	TCP	53	>1023	Replies using TCP (Cisco routers: allow establish)
big3d	LDNS	UDP	53	33434	Probe using UDP or traceroute utility
LDNS	big3d	ICMP	N/A	N/A	Replies to ICMP, UDP pings, or traceroute probes
big3d	LDNS	dns_rev dns_dot	>1023	53	Probe using DNS version or DNS dot
LDNS	big3d	dns_rev dns_dot	53	>1023	Replies to DNS version or DNS dot probes

Table 4.7 Communications between **big3d** agents and local DNS servers



5

DNS Resource Records

- Understanding DNS resource records
- Types of resource records
- Additional resource record types

Understanding DNS resource records

A *resource record* is an entry in a DNS database file and consists of a name, a TTL, a type, and data that is specific to the type. These resource records, in a hierarchical structure, make up the domain name system (DNS).

The standard resource record format, specified in RFC 1035, is as follows:

```
{name} {ttl} addr-class record type record-specific data
```

The resource record fields are defined as follows:

◆ **name**

The first field, **name**, is the name of the domain record and it must always start in column 1. For all resource records that are not the first in a file, the name may be left blank. When the name field is left blank, the record takes name of the previous resource record.

◆ **ttl**

The second field, **ttl** (time to live), is optional. This field specifies how long the resource record is stored by the LDNS. If this field is left blank, the default TTL value is specified in the start of authority (SOA) resource record (described later in this chapter).

◆ **address class**

The third field is the address class. Currently, only one class is supported: **IN**, for internet addresses and other internet information. Limited support is included for the **HS** class, which is for MIT/Athena "Hesiod" information.

◆ **record type**

The fourth field, **record type**, defines the type of this resource record, such as **A**, **NS**, or **CNAME**.

◆ **other fields**

Additional fields may be present in a resource record, depending on its type.

Although case is preserved in names and data fields when loaded into the name server, comparisons and lookups in the name server database are not case-sensitive.

◆ **Note**

*For more information about resource records, DNS, and related topics, refer to **DNS and BIND**, by Albitz and Liu.*

Types of resource records

Many types of resource records are currently in use. This section provides an overview of the most common resource record types, and lists other types of resource records. The six most common types of resource records are shown in Table 5.1.

Resource Record Type	Description
A (Address)	Maps host names to IP addresses.
CNAME (Canonical Name)	Defines a host alias.
MX (Mail Exchange)	Identifies where to send mail for a given domain name.
NS (Name Server)	Identifies the name servers for a domain.
PTR (Pointer)	Maps IP addresses to host names.
SOA (Start of Authority)	Indicates that a name server is the best source of information for a zone's data; defines the default parameters for a zone.

Table 5.1 Common resource records

A (Address)

The Address record, or **A** record, lists the IP address for a given host name. The name field is the host's name, and the address is the network interface address. There should be one **A** record for each IP address of the machine.

Figure 5.1 shows an example of an **A** record.

{name}	{ttl}	addr-class	{type}	address
host1.domain.com		IN	A	128.32.0.4
		IN	A	10.0.0.78

Figure 5.1 Example of an A record

CNAME (Canonical Name)

The Canonical Name resource record, **CNAME**, specifies an alias or nickname for the official, or canonical, host name. This record must be the only one associated with the alias name. It is usually easier to supply one **A** record for a given address and use **CNAME** records to define alias host names for that address.

Figure 5.2 shows an example of a **CNAME** resource record:

alias	{ttl}	addr-class	{type}	Canonical name
wip.domain.com		IN	CNAME	host1.domain.com

Figure 5.2 Example of a CNAME record

MX (Mail Exchange)

The Mail Exchange resource record, **MX**, defines the mail system(s) for a given domain.

Figure 5.3 shows an example of an **MX** resource record.

name	{ttl}	addr-class	MX	pref value	mail exchange
Munnari.OZ.AU.		IN	MX	0	Seismo.CSS.GOV.
*.IL.		IN	MX	0	RELAY.CS.NET.

Figure 5.3 Example of an MX record

NS (Name Server)

The name server resource record, **NS**, defines the name servers for a given domain, creating a delegation point and a subzone. The first **name** field specifies the zone that is served by the name server that is specified in the **name servers name** field. Every zone needs at least two name servers.

Figure 5.4 shows an example of an **NS** resource record.

{name}	{ttl}	addr-class	NS	Name servers name
domain.com		IN	NS	host1.domain.com.
domain.com		IN	NS	host2.domain.com.

Figure 5.4 Example of an NS record

PTR (Pointer)

A name pointer resource record, **PTR**, associates a host name with a given IP address. These records are used for reverse name lookups.

The example of a **PTR** record shown in Figure 5.5 is used to set up reverse pointers for the special **IN-ADDR.ARPA** domain.

name	{ttl}	addr-class	PTR	real name
7.0		IN	PTR	monet.Berkeley.Edu.

Figure 5.5 Example of a PTR record

SOA (Start of Authority)

The start of authority resource record, **SOA**, starts every zone file and indicates that a name server is the best source of information for a particular zone. In other words, the **SOA** record indicates that a name server is authoritative for a zone. There must be exactly one **SOA** record per zone.

The following is an example of an **SOA** record.

name	{ttl}	addr-class	SOA	Origin	Person in charge
@		IN	SOA	ucbvax.Berkeley.Edu.	john DOE.berkeley.edu (
				1995122103	; Serial
				10800	; Refresh
				1800	; Retry
				3600000	; Expire
				259200)	; Minimum

Figure 5.6 Example of an SOA record

The specific fields in an **SOA** record are defined as follows:

- ◆ **Person in charge**
The email address for the person responsible for the name server, with the at character (@) changed to a dot (.). For example, **john DOE@berkeley.edu** becomes **john DOE.berkeley.edu**.
- ◆ **Serial number**
The version number of the data file; it must be a positive integer. You must increase this number whenever a change is made to the data.
- ◆ **Refresh**
The time interval between calls, in seconds, that the secondary name servers make to the primary name server to check if an update is necessary.
- ◆ **Retry**
The time interval, in seconds, that a secondary server waits before retrying a failed zone transfer.
- ◆ **Expire**
The maximum number of seconds that a secondary name server can use the data before it expires for lack of receiving a refresh.
- ◆ **Minimum**
The default number of seconds to be used for the time to live (TTL) field on resource records which do not specify a TTL in the zone file. It is also an enforced minimum on TTL if it is specified on a resource record in the zone.

Additional resource record types

Table 5.2 lists less common resource record types. For more information on these, see RFCs 1035, 1183, and 1664.

Resource Record Type	Description
AAAA	IPv6 address
AFSDB	AFS database location
GPOS	Geographical position
HINFO	Host information
ISDN	Integrated services digital network address
KEY	Public key
KX	Key exchanger
LOC	Location information
MB	Mailbox domain name
MINFO	Mailbox or mail list information
NULL	A null RR
NSAP	Network service access point address
NSAP-PTR	(Obsolete)
NXT	Next domain
PX	Pointer to X.400/RFC822 information
RP	Responsible person
RT	Route through
SIG	Cryptographic signature
SRV	Server selection
TXT	Text strings
WKS	Well-known service description
X25	X25

Table 5.2 Less common resource records



6

Extended Content Verification (ECV)

- Working with the ECV Service Monitor

Working with the ECV Service Monitor

When you set up an extended content verification (ECV) service monitor for a wide IP, you can monitor not only the availability of a port or service on a server, but also the availability of a specific file on a particular server. An ECV service monitor verifies whether a specific file is available using the HTTP, HTTPS, or FTP network services. You can also specify a search string for the ECV monitor. When you specify a search string, the 3-DNS not only verifies that a file is available, but also that whatever you specify in the search string is in the file.

An ECV service monitor can help you ensure that clients are getting what they are after, and that they will not get an error, whether they are looking for information, making an online purchase, or uploading software.

An ECV service monitor works in the following manner: if the file responds appropriately to the ECV query, the server where the file resides is marked as **up** and the client connection request is sent to that server. If the file does not respond as expected to the ECV query, the server where the file resides is marked as **down**, and the client will not be sent to that server.

Defining ECV service monitors

You can define ECV service monitors using the Configuration utility, or from the command line. You define ECV monitors for wide IPs only.

To define ECV service monitors using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
The Wide IP List screen opens.
2. In the Wide IP column, click the wide IP to which you want to add an ECV service monitor.
The Modify Wide IP screen opens.
3. Add the settings for the ECV near the bottom of the screen, and click **Update**. For more information on the ECV settings, click **Help** on the toolbar.

To define ECV service monitors from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.

3. Use the syntax shown in Figure 6.1 to define an ECV service monitor.

You should place all ECV service monitor statements just before the wide IP pool definitions in the **wideip.conf** file.

◆ **Note**

*You can set up ECV monitors that use the **https** protocol only on crypto 3-DNS systems.*

```
ecv {
  protocol      <http | https | ftp>
  filename      <"path and file name">
  scan_level    <all | first>
  user          [ <"user name"> ]
  hashed_password [ <"hashed version of server password"> ]
  search_string <"search string text">
}
```

Figure 6.1 Syntax for defining ECV service monitors

Figure 6.2 shows a sample ECV statement that defines an ECV service monitor in the **wideip.conf** file.

```
ecv {
  protocol      http
  filename      "/home/user/readme.txt"
  scan_level    all
  user          "jones"
  hashed_password "22AECCCD9CA9C2CC8B"
  search_string "Configuration Notes"
}
```

Figure 6.2 Sample ECV service monitor definition

Using the search string option

With the search string option, you can specify text or characters that you want the ECV monitor to verify within the file. The search string functionality is based on POSIX regular expression matching. Regular expressions are a matching tool for text and characters within a file. When you include a search string in an ECV service monitor, the 3-DNS not only verifies that the file exists, but also that whatever text you type for the search string is available, exactly as you typed it, within the file. The most basic search string options are simply text. For a more advanced search string, you can use the POSIX regular expression characters shown in Table 6.1.

Character	Description
^	Specifies the start of a line.
\$	Specifies the end of a line.
*	Specifies any number of characters up to the end of a line or a file.
?	Specifies one instance of any character.
\	Releases any regular expression interpretation of the following character.
!	Implies that if the string is not found, the wide IP status is up . Use at the beginning of the search string.

Table 6.1 POSIX regular expression characters for ECV search strings

◆ **Note**

*For more information on working with POSIX regular expressions, refer to the **re_format** man page. To view the **re_format** man page, type the following at the command line:*

```
man re_format
```




7

Internet Weather Map

- Overview of the Internet Weather Map
- Interpreting the Internet Weather Map data

Overview of the Internet Weather Map

The Internet Weather Map statistics screen, in the Configuration utility, provides the following real-time data about the Internet:

- The average round trip time between the local DNS servers on a particular continent and the data centers in your network
- The average completion rate between the local DNS servers on a particular continent and the data centers in your network
- The average number of router hops between the local DNS servers on a particular continent and the data centers in your network

The data displayed in the Internet Weather Map is based on path data, which is collected when you use a dynamic load balancing mode such as Round Trip Times or Quality of Service. For more information on dynamic load balancing modes, see *Using dynamic load balancing modes*, on page 8-5.

To view the Internet Weather Map statistics screen using the Configuration utility

1. Expand the **Statistics** item in the navigation pane.
2. Click **Weather Map**.
The Internet Weather Map Statistics screen opens.
3. For information on working with the Internet Weather Map Statistics screen, view the online help.

◆ Note

The Internet Weather Map requires the IP classifier, which is not available on the no-crypto 3-DNS. If you are using a no-crypto 3-DNS, the Internet Weather Map shows no valid data.

The round trip time and completion rate data on the Internet Weather Map Statistics screen is based on path data. If you do not have path probing activated, the data on this screen will be stale. The router hops data is based on information collected by the **traceroute** utility. If you do not allow the 3-DNS to collect hops information, the average router hops data will be stale.

To activate path probing and hops data collection using the Configuration utility


1. Click **System** in the navigation pane.
The System - General screen opens.
2. Click **Metric Collection** on the toolbar.
The System - Metric Collection screen opens.
3. Check the **Allow Probing** box.
The 3-DNS can now collect path information for the data centers in your configuration.

4. Check the **Allow Hops** box.
The 3-DNS can now collect router hops information for the data centers in your configuration.

Working with the Average Round Trip Time table

In the Average Round Trip Time table on the Internet Weather Map Statistics screen, you can view the following information:

- The average round trip time for each data center to each continent
- For each data center, the best average round trip time to the local DNS servers on a particular continent. This value is indicated by **bold** text within the table.
- For each continent, the best average round trip time from the data centers. This value is indicated by underlined text within the table.


If you hold the mouse pointer over the Information button (), you can view the following additional information:

- For a particular data center, the number of local DNS servers used to calculate the average round trip time
- For all the local DNS servers that have been probed by a particular data center, the percentage of those local DNS servers that are located on a particular continent
- For all the local DNS servers on a particular continent, the percentage of those local DNS servers that have been probed by a particular data center

Working with the Average Completion Rate table

In the Average Completion Rate table on the Internet Weather Map Statistics screen, you can view the following information:

- The average completion rate for each data center to each continent
- For each data center, the best average completion rate to the local DNS servers on a particular continent. This value is indicated by **bold** text within the table.
- For each continent, the best average completion rate from the data centers. This value is indicated by underlined text within the table.


If you hold the mouse pointer over the Information button (), you can view the following additional information:

- For a particular data center, the number of local DNS servers used to calculate the average completion rate
- For all the local DNS servers that have been probed by a particular data center, the percentage of those local DNS servers that are located on a particular continent
- For all the local DNS servers on a particular continent, the percentage of those local DNS servers that have been probed by a particular data center

Working with the Average Router Hops table

In the Average Router Hops table on the Internet Weather Map Statistics screen, you can view the following information:

- The average number of router hops between each data center and each continent
- For each data center, the best average number of router hops to the local DNS servers on a particular continent. This value is indicated by **bold** text within the table.
- For each continent, the best average number of router hops from the data centers. This value is indicated by underlined text within the table.

If you hold the mouse pointer over the Information button (), you can view the following additional information:

- For a particular data center, the number of local DNS servers used to calculate the average number of router hops
- For all the local DNS servers that have been probed by a particular data center, the percentage of those local DNS servers that are located on a particular continent
- For all the local DNS servers on a particular continent, the percentage of those local DNS servers that have been probed by a particular data center

Interpreting the Internet Weather Map data

You can use the data in the Internet Weather Map (IWM) to compare performance between data centers. By comparing data center performance over time, you can stage your content in the data centers based on actual usage. The two data points that help you determine which data center has the best performance are the RTT response time (lower is better), and the Completion Rate (higher is better). One easy way to compare data center performance over time is to print a screen shot of the IWM at a certain time every day.

You can also use the IWM data to determine which data center best serves content for which continent. By analyzing which data center provides the best response (usually the lowest RTT and the highest relative completion rate) for a given continent, you can localize your content in the data center that provides the most efficient content delivery.



8

Load Balancing

- Working with load balancing modes
- Understanding load balancing on the 3-DNS
- Configuring load balancing
- Changing global variables that affect load balancing
- Setting up load balancing for services that require multiple ports
- Troubleshooting manual configuration problems

Working with load balancing modes

The 3-DNS uses load balancing modes to distribute DNS name resolution requests, sent by local DNS servers, to the best available virtual server in your network. This chapter first describes how load balancing works on the 3-DNS, explains the various static and dynamic load balancing modes, and then describes how to configure them.

Understanding load balancing on the 3-DNS

When the 3-DNS receives a name resolution request from a local DNS server, the system uses a load balancing mode to select the best available virtual server from a wide IP pool. Once the 3-DNS selects the virtual server, it constructs the DNS answer and sends the answer back to the requesting client's local DNS server. The DNS answer, or *resource record*, can be either an **A** record that contains virtual server IP addresses, or a **CNAME** record that contains the canonical name for a DNS zone.

The 3-DNS chooses a virtual server from a wide IP pool using either a *static load balancing mode*, which selects a virtual server based on a pre-defined pattern, or a *dynamic load balancing mode*, which selects a virtual server based on current performance metrics.

The 3-DNS uses load balancing modes in two situations:

- ◆ **Load balancing among multiple pools**

The 3-DNS supports multiple pools. Configurations that contain two or more pools use a load balancing mode first to select a pool. Once the 3-DNS selects a pool, the system then uses a load balancing mode to choose a virtual server within the selected pool. If the 3-DNS does not choose a virtual server in the first pool, it applies the load balancing mode to the next pool, either until it selects the best virtual server to respond to the request, or all the pools are tried.

- ◆ **Load balancing within a pool**

Within each pool, you specify three different load balancing modes that the system uses in sequential order: preferred method, alternate method, and fallback method. The *preferred* method is the first load balancing mode that the 3-DNS uses for load balancing. If the preferred method fails, the system then uses the alternate method for load balancing. If this load balancing mode fails, the system uses the fallback load balancing mode. If the fallback method fails, the 3-DNS returns the client to standard DNS for resolution.

Table 8.1 shows a complete list of the supported load balancing modes, and indicates where you can use each mode in the 3-DNS configuration. The following sections in this chapter describe how each load balancing mode works.

Load Balancing mode	Use for pool load balancing	Use for preferred method	Use for alternate method	Use for fallback method
Completion Rate		X		X
Global Availability	X	X	X	X
Hops		X		X
Kilobytes/Second		X		X
Least Connections		X		X
None		X	X	X
Packet Rate		X	X	X
Quality of Service		X		X
Random	X	X	X	X
Ratio	X	X	X	X
Return to DNS		X	X	X
Round Robin	X	X	X	X
Round Trip Time		X		X
Static Persist		X	X	X
Topology	X	X	X	X
VS Capacity		X	X	X

Table 8.1 Load balancing mode usage

Using static load balancing modes

Static load balancing modes distribute connections across the network according to predefined patterns, and take server availability into account. The 3-DNS supports the following static load balancing modes:

- Global Availability
- None
- Random
- Ratio

- Return to DNS
- Round Robin
- Static Persist
- Topology

The None and Return to DNS load balancing modes are special modes that you can use to skip load balancing under certain conditions. The other static load balancing modes perform true load balancing as described in the following sections.

Global Availability mode

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

None mode

The None load balancing mode is a special mode you can use if you want to skip the current load balancing method, or skip to the next pool in a multiple pool configuration. For example, if you set an alternate method to None in a pool, the 3-DNS skips the alternate method and immediately tries the load balancing mode specified as the fallback method. If the fallback method is set to None, and you have multiple pools configured, the 3-DNS uses the next available pool. If you do not have multiple pools configured, the 3-DNS returns the connection request to DNS for resolution.

This mode is most useful for multiple pool configurations. For example, you can temporarily remove a specific pool from service by setting each of the methods (preferred, alternate, and fallback) to None. (Note that you can also disable a pool from the Modify Wide IP Pools screen, in the Configuration utility.) You could also use the mode to limit each pool to a single load balancing mode. For example, you would set the preferred method in each pool to the desired load balancing mode, and then you would set both the alternate and fallback methods to None in each pool. If the preferred method fails, the None mode in both the alternate and fallback methods forces the 3-DNS to go to the next pool for a load balancing answer.

Random mode

The Random load balancing mode sends connections to virtual servers in a random, uniform distribution pattern. The Random mode is useful for certain test configurations.

Ratio mode

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

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

Figure 8.1 shows a sample connection distribution for Ratio mode.



Figure 8.1 Ratio mode

Return to DNS mode

The Return to DNS mode is another special load balancing mode that you can use to immediately return connection requests to DNS for resolution. This mode is particularly useful if you want to temporarily remove a pool from service, or if you want to limit a pool in a single pool configuration to only one or two load balancing attempts.

Round Robin mode

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

Figure 8.2 shows a sample of the connection distribution pattern for Round Robin mode.



Figure 8.2 Round Robin mode

Static Persist mode

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

Topology mode

The Topology load balancing mode allows you to direct or restrict traffic flow by adding topology records to a topology statement in the configuration file. When you use the Topology load balancing mode, you can develop proximity-based load balancing. For example, a client request in a particular geographic region can be directed to a data center or server within that same region. The 3-DNS determines the proximity of servers by comparing location information derived from the DNS message to the topology records.

This load balancing mode requires you to do some advanced configuration planning, such as gathering the information you need to define the topology records. The 3-DNS contains an IP classifier that accurately maps local DNS servers, so when you create topology records, you can refer to continents and countries, instead of IP subnets.

See Chapter 13, *Topology*, for detailed information about working with this and other topology features. For an example configuration using the Topology load balancing mode, see the *3-DNS Administrator Guide*, Chapter 7, *Configuring a Globally-Distributed Network*.

Using dynamic load balancing modes

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

All dynamic load balancing modes make load balancing decisions based on the metrics collected by the **big3d** agents running in each data center. The **big3d** agents collect the information at set intervals that you define when you set the global timer variables. If you want to use the dynamic load balancing modes, you must run one or more **big3d** agents in each of your data centers, to collect the required metrics.

The 3-DNS supports the following dynamic load balancing modes:

- Completion Rate
- Hops
- Kilobytes/Second
- Least Connections
- Packet Rate

- Round Trip Times (RTT)
- Quality of Service (QOS)
- VS Capacity

Completion Rate mode

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

Figure 8.3 shows a sample connection distribution pattern for the Completion Rate mode.

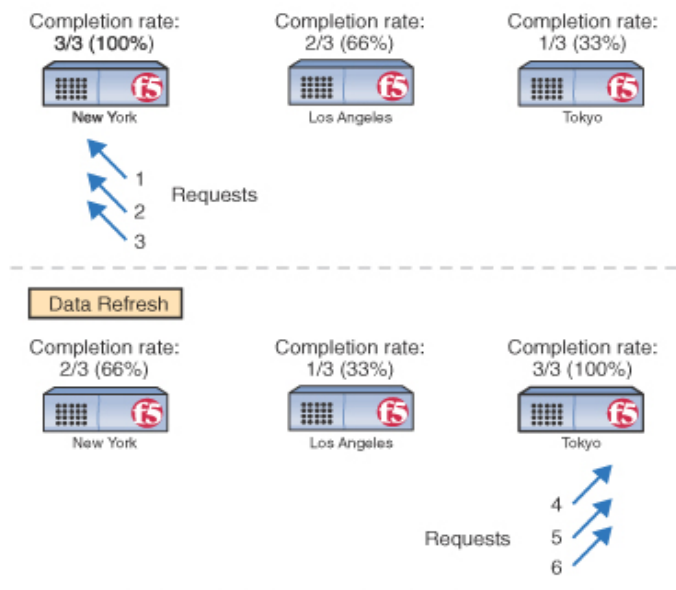


Figure 8.3 Completion Rate load balancing mode

Hops mode

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

Kilobyte/Second mode

The Kilobytes/Second load balancing mode selects a virtual server that is currently processing the fewest number of kilobytes per second. Note that you can use the Kilobytes/Second mode only with servers for which the

3-DNS can collect the kilobytes per second metric. See *Configuring host SNMP settings on the 3-DNS*, on page 12-7, for details on the metrics the 3-DNS collects.

Least Connections mode

The Least Connections load balancing mode is used for load balancing to virtual servers managed by BIG-IP systems. The Least Connections mode simply selects a virtual server on the BIG-IP that currently hosts the fewest connections.

Packet Rate mode

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

Figure 8.4 shows a sample connection distribution for the Packet Rate mode.

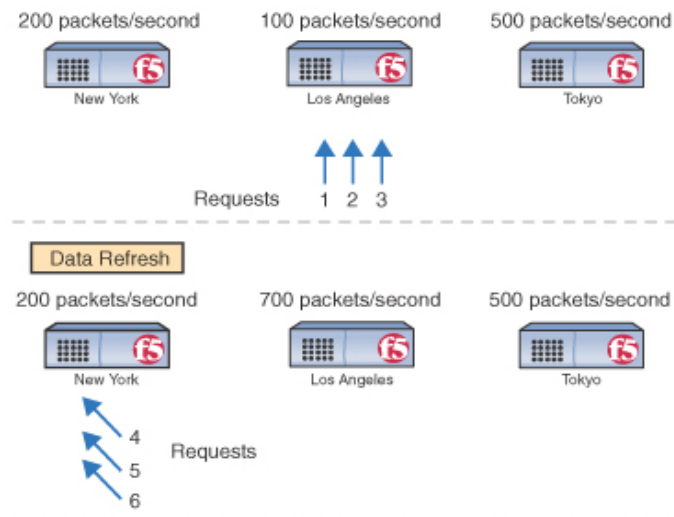


Figure 8.4 Packet Rate mode

Round Trip Times mode

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

Figure 8.5 shows a sample connection distribution for the Round Trip Times mode.

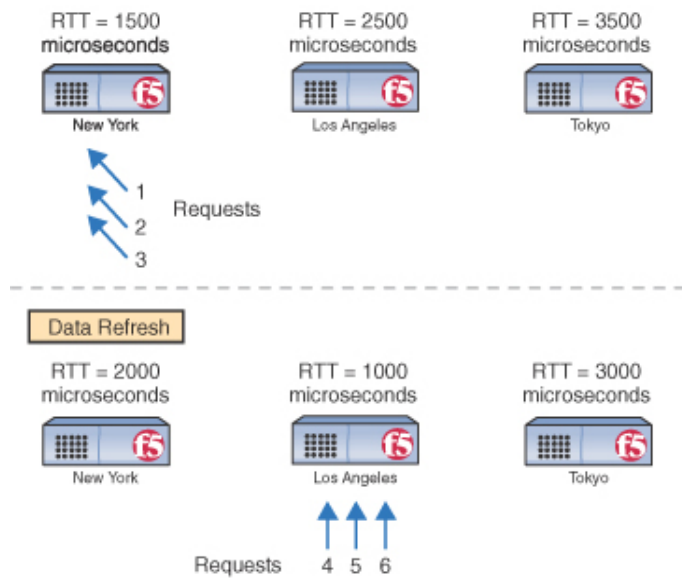


Figure 8.5 Round Trip Times mode

Quality of Service mode

The Quality of Service load balancing mode uses current performance information to calculate an overall score for each virtual server, and then distributes connections based on each virtual server's score. The performance factors that the 3-DNS takes into account include:

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

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

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

For details about customizing the Quality of Service mode, see the *3-DNS Administrator Guide*, Chapter 9, *Working with Quality of Service*.

VS Capacity mode

The VS Capacity load balancing mode creates a list of the virtual servers, weighted by capacity, then picks one of the virtual servers from the list. The virtual servers with the greatest capacity are picked most often, but over time all virtual servers are returned. If more than one virtual server has the same capacity, then the 3-DNS load balances using the Random mode among those virtual servers.

In the sample configuration in Figure 8.6, **VS 1** would be chosen three times as often as **VS 3**, and 2/3 as often as **VS 2**. **VS 2** would be chosen twice as often as **VS 3**. If one of the nodes behind **VS 1** became unavailable, then **VS 1** and **VS 2** would be chosen with about the same frequency, but twice as often as **VS 3**.

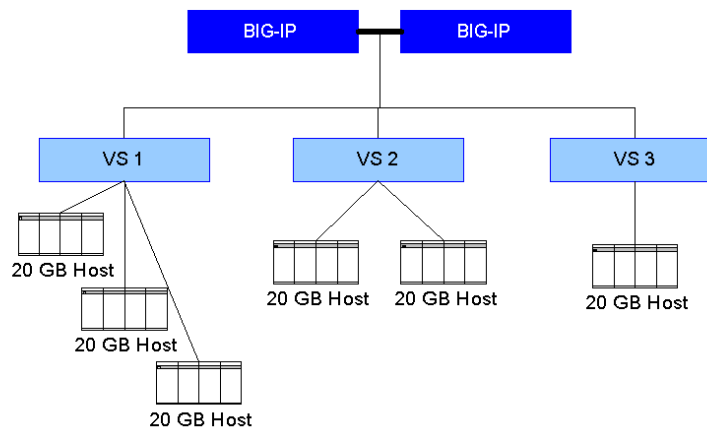


Figure 8.6 VS Capacity load balancing mode

Configuring load balancing

This section describes how to configure load balancing on the 3-DNS. You configure load balancing at the global, wide IP, and pool levels:

◆ Global

At the global level, you can configure default settings for the alternate and fallback load balancing methods. Then, if you do not specify alternate or fallback modes when defining a wide IP, the 3-DNS uses the alternate and fallback methods you have configured at the global level. You can find instructions on how to configure global alternate and fallback methods in *Setting global alternate and fallback methods*, on page 8-16.

◆ **Wide IP**

When you define a wide IP, and you have multiple pools in your wide IP, you first specify which load balancing mode to use in selecting a pool in the wide IP. Next, you specify which preferred, alternate, and fallback load balancing methods to use in selecting a virtual server within the selected pool. You can find instructions on how to configure these load balancing methods in the section, *Defining a wide IP*, on page 8-11.

Understanding wide IPs

After you configure the BIG-IP systems, EDGE-FX Caches, hosts, and the virtual servers they manage, you need to group the configured virtual servers into wide IPs. A **wide IP** is a mapping of a fully-qualified domain name (FQDN) to a set of virtual servers that host the domain's content, such as a web site, an e-commerce site, or a CDN.

Before defining the first wide IP, you should do the following:

- ◆ Gather your configuration information for the BIG-IP, EDGE-FX Cache, and host so you can easily see which virtual servers have the content you want to map to an FQDN. Then you can decide how to group virtual servers into pools.
- ◆ Decide which load balancing modes to use for each pool of virtual servers.

◆ **Note**

*When you run the 3-DNS in node mode, NameSurfer, a third-party application included with the 3-DNS, sets up DNS zone files so that wide IP definitions are properly linked to DNS. NameSurfer registers the virtual servers you add to wide IP pools as A records. No action is required on your part, as NameSurfer automatically handles this process. For more information on NameSurfer, see the online help that is included with the application. (To view the NameSurfer application, click **NameSurfer** in the navigation pane).*

There may be situations (for example, e-commerce, and other sites with multiple services) where you need to configure a wide IP so that connections are not sent to a given address unless multiple ports or services are available. You configure this behavior after you define the wide IP. For details, see *An example configuration using a port list*, on page 8-21.

Understanding pools

A wide IP contains one or more pool definitions. A **pool** is a group of virtual servers that the 3-DNS load balances. You can include all types of virtual servers (BIG-IP, EDGE-FX Cache, and host) in a pool definition.

Defining a wide IP

After you determine which virtual servers you should place in which wide IP pools, you are ready to add the first wide IP to the configuration. Note that you must configure at least one pool in the wide IP, but you may configure any number of pools.

To define a wide IP using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
The Wide IP List screen opens.
2. On the toolbar, click **Add Wide IP**.
The Add a New Wide IP screen opens.
3. Add the wide IP settings, and click **Next**.
The Configure Load Balancing for New Pool screen opens.
4. Add the pool settings, and click **Next**.
The Select Virtual Servers screen opens.
5. Check the virtual servers that you want to add to the pool, and click **Finish**.
The wide IP is added to your configuration.

Repeat this process for each wide IP you want to add. For help on defining wide IPs and pools, click **Help** on the toolbar.

To define a wide IP from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Add a **wideip** statement.
Place the **wideip** statement after all **server** statements and before any **topology** statement.
4. Under the **wideip** statement, enter the wide IP address, port, and name information. Enclose the wide IP name in quotation marks.
5. Configure any options you want to set (such as the TTL, port list, or QOS coefficients) by entering the appropriate sub-statements.
6. Define the **pool** sub-statement. At the minimum, the **pool** sub-statement should include its name (enclosed in quotation marks) and the virtual servers it contains.
7. Define the load balancing modes you want to use by entering **preferred**, **alternate**, and **fallback** sub-statements.
8. Define the IP address, port, and ratio value for each virtual server that you want to include in this pool.

Figure 8.7 shows the correct syntax for the **wideip** statement.

```

wideip {
  address <ip_addr>
  port <port_number> | <"service name">
  persist < yes | no >
  persist_ttl <number>
  name <"domain_name">
  [ alias <"alias_name"> ]
  [ ttl <number> ]
  [ port_list <port_number> <port_number> ... ]
  [ qos_coeff {
    rtt <n>
    completion_rate <n>
    packet_rate <n>
    topology <n>
    hops <n>
    vs_capacity <n>
    kbps <n>
  } ]
  [ pool_lbmode <rr | ratio | ga | random | topology> ]
  pool {
    name <"pool_name">
    [ limit {
      kbytes_per_second
      pkts_per_second <number>
      current_conns <number>
      cpu_usage <number>
      mem_avail <number>
      disk_avail <number>
    } ]
    [ ratio <pool_ratio> ]
    [ dynamic_ratio < yes | no > ]
    [ rr_ldns < yes | no > ]
    [ preferred < completion_rate | ga | hops | kbps | leastconn | packet_rate | qos |
random | ratio | return_to_dns | rr | rtt | topology | null | vs_capacity |
static_persist> ]
    [ alternate < ga | null | random | ratio | return_to_dns | rr | topology |
vs_capacity | static_persist> ]
    [ fallback <completion_rate | ga | hops | leastconn | null | packet_rate | qos |
random | ratio | return_to_dns | rr | rtt | topology | vs_capacity | static_persist> ]
    address <vs_addr>[:<port>] [ratio <weight>]
    address <vs_addr>[:<port>] [ratio <weight>]
    address <vs_addr>[:<port>] [ratio <weight>]
    ...
  }
}

```

Figure 8.7 Syntax for the **wideip** statement

Using wildcard characters in wide IP names

The 3-DNS supports wildcard characters in wide IP names and wide IP aliases. You can use the wildcard characters to simplify your maintenance tasks if you have a large quantity of wide IP names and/or wide IP aliases. The wildcard characters you can use are: the question mark (?), and the asterisk (*). The guidelines for using the wildcard characters are as follows:

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

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

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

◆ Note

There are two important things to keep in mind when you use wildcard characters. First, wildcard characters are not inserted into NameSurfer. Second, if you are using ECV service monitors, they do not scan wide IP names or aliases that contain wildcard characters.

An example of the wideip statement

Figure 8.8 shows a sample **wideip** statement. This statement defines a wide IP named **mx.wip.domain.com**, with an alias of **mail.wip.domain.com**. The wide IP contains two pools, with **pool_1** receiving three times as many requests as **pool_2**. The 3-DNS attempts to resolve requests sent to **pool_1** using the Round Trip Times (RTT) mode. This mode sends connections to the virtual server in the pool that demonstrates the best round trip time between the virtual server and the client LDNS. If the 3-DNS cannot resolve the request using the RTT mode, the system distributes requests using the

Random load balancing mode. The 3-DNS distributes requests at a 2:1 ratio to the two virtual servers defined in **pool_2**, where the first listed virtual server receives twice as many connections as the second.

```
wideip {
  address      192.168.102.50
  service      "smtp"
  name         "mx.wip.domain.com"
  alias        "mail.wip.domain.com"
  pool_lbmode  ratio
  pool {
    name       "pool_1"
    ratio      3
    preferred  rtt
    alternate  random
    address    192.168.101.50
    address    192.168.102.50
    address    192.168.103.50
  }
  pool {
    name       "pool_2"
    ratio      1
    preferred  ratio
    address    192.168.104.50  ratio 2
    address    192.168.105.50  ratio 1
  }
}
```

Figure 8.8 Example syntax for defining a wide IP

Using the LDNS round robin wide IP attribute

LDNS round robin is an attribute that you can use in conjunction with any load balancing mode. The LDNS round robin attribute allows the 3-DNS to return a list of available virtual servers, instead of a single virtual server. Certain browsers keep the answer returned by DNS servers. By enabling this attribute, the 3-DNS returns a maximum of 16 virtual servers as the answer to a DNS resolution request. This provides browsers with alternate answers if a virtual server becomes unavailable.

Using the last resort pool designation

The last resort pool is an optional setting for a wide IP pool. The wide IP pool that you designate as the last resort pool, in the Configure Load Balancing for New Pool screen, is the virtual server pool that the 3-DNS uses when all other pools have reached their thresholds or are unavailable for any reason. The 3-DNS uses the last resort pool only when it tries, unsuccessfully, to load balance to all other configured pools

When your network includes cache appliances hosting content from an origin site, you can designate the origin site as the last resort pool to handle requests if your cache virtual servers have reached their thresholds. You can

also use the last resort pool to designate an overflow network so your origin servers remain available if network traffic spikes. You can only designate one last resort pool within a wide IP.

To designate a last resort pool using the Configuration utility

1. In the navigation pane, select **Wide IPs**.
The Wide IP List screen opens.
2. From the Pools column, select the pools for the wide IP for which you want to create a last resort pool.
The Modify Wide IP Pools screen opens.
3. From the Pool Name column, click the pool that you want to designate as the last resort pool.
The Modify Load Balancing for [pool name] screen opens.
4. Check the box next to **Last Resort Pool**, and click **Update**.

To designate a last resort pool from the command line

In the `wideip.conf` file, change the `last_resort` definition from `no` to `yes` for the pool that you want to designate as the last resort pool. Figure 8.9 shows an example of a last resort pool definition.

```
pool {
  name "origin"
  last_resort yes
  preferred kbps
  alternate rr
  fallback return_to_dns
  address 192.168.103.5
  address 192.168.103.6
  address 192.168.103.7
}
```

Figure 8.9 Example of a last resort pool definition

Changing global variables that affect load balancing

You can configure global variables that affect how load balancing is handled on a global basis for all wide IPs managed by the 3-DNS. You can override these global settings for individual wide IPs as necessary.

Global variables that affect load balancing fall into two categories:

- Alternate and fallback load balancing methods
- TTL (time to live) and timer values

The default settings for these variables are adequate for most configurations. However, if you want to change any global variable, you should refer to the online help.

Setting global alternate and fallback methods

You can configure a load balancing method that all wide IPs can use in the event that their preferred method fails.

To configure global alternate and fallback load balancing methods using the Configuration utility

1. In the navigation pane, click **System**.
The System - General screen opens.
2. On the toolbar, click **Load Balancing**.
3. In the **Default Alternate** box, select the load balancing mode to use should a wide IP's preferred method fail.
4. In the **Default Fallback** box, specify the load balancing mode to use should the preferred and alternate methods fail.
If all methods fail, requests are returned to DNS for resolution.
5. Finish configuring the rest of the settings on the System - Load Balancing screen. (For help on configuring the load balancing settings, click **Help** on the toolbar.)
The global load balancing settings are added to your configuration.

To configure global alternate and fallback load balancing methods from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate or add the **globals** statement. The **globals** statement should be at the top of the file.
4. Use the syntax shown in Figure 8.10 to define global alternate and fallback load balancing methods.

```
globals {
  [ default_alternate < ga | leastconn | null | packet_rate | random | ratio |
  return_to_dns | rr | topology | static_persist | vs_capacity > ]
  [ default_fallback < completion_rate | ga | hops | leastconn | null | packet_rate |
  qos | random | ratio | return_to_dns | rr | rtt | topology | static_persist |
  vs_capacity> ]
}
```

Figure 8.10 Configuring global alternate and fallback load balancing modes

Figure 8.11 shows a sample **globals** statement that defines global load balancing variables.

```
globals {
  default_alternate leastconn
  default_fallback rr
}
```

Figure 8.11 Sample syntax for setting global load balancing variables

Understanding TTL and timer values

Each 3-DNS object has an associated *time-to-live (TTL)* value. A TTL is the amount of time (measured in seconds) for which metrics information is considered valid. The timer values determine how often the 3-DNS refreshes the information.

Table 8.2 describes each TTL value, as well as its default setting.

Parameter	Description	Default
Server TTL	Specifies the number of seconds that the 3-DNS uses BIG-IP and EDGE-FX Cache metrics information for name resolution and load balancing.	60
Host TTL	Specifies the number of seconds that the 3-DNS uses generic host machine metrics information for name resolution and load balancing.	240
3-DNS TTL	Specifies the number of seconds that the 3-DNS considers performance data for the other 3-DNS systems to be valid.	60
Virtual server TTL	Specifies the number of seconds that the 3-DNS uses virtual server information (data acquired from a BIG-IP, EDGE-FX Cache, or host about a virtual server) for name resolution and load balancing.	120
Hops TTL	Specifies the number of seconds that the 3-DNS considers traceroute data to be valid.	604800 (seven days)
Path TTL	Specifies the number of seconds that the 3-DNS uses path information for name resolution and load balancing.	2400
Default TTL	Specifies the default number of seconds that the 3-DNS considers a wide IP A record to be valid. If you do not specify a wide IP TTL value when defining a wide IP, the wide IP definition uses the default_ttl value.	30

Table 8.2 TTL values and default settings

Each 3-DNS object also has a timer value. A timer value defines the frequency (measured in seconds) at which the 3-DNS refreshes the metrics information it collects. In most cases, the default values for the TTL and

timer parameters are adequate. However, if you make changes to any TTL or timer value, keep in mind that an object's TTL value must be greater than its timer value.

Table 8.3 describes each timer value, as well as its default setting.

Parameter	Description	Default
Server data refresh	Specifies the frequency (in seconds) at which the 3-DNS refreshes BIG-IP and EDGE-FX Cache information.	20
Host data refresh	Specifies the frequency (in seconds) at which the 3-DNS refreshes other host machine information.	90
3-DNS data refresh	Specifies the frequency (in seconds) at which the 3-DNS retrieves performance data for other 3-DNS systems in the sync group.	20
Virtual server data refresh	Specifies the frequency (in seconds) at which the 3-DNS refreshes virtual server information.	30
ECV timer refresh	Specifies the frequency (in seconds) at which the 3-DNS refreshes the ECV monitor.	90
Hops data refresh	Specifies the frequency (in seconds) at which the 3-DNS retrieves traceroute data (traceroutes between each data center and each local DNS).	60
Path data refresh	Specifies the frequency (in seconds) at which the 3-DNS refreshes path information (for example, round trip time or ping packet completion rate).	120
Remote nodes query	Specifies the frequency (in seconds) at which the 3-DNS queries remote 3-DNS systems and BIG-IP systems.	60
3-DNS Sync Time Tolerance	Specifies the number of seconds that one system's time setting is allowed to be out of sync with another system's time setting. Note: If you are using NTP to synchronize the time of the 3-DNS with a time server, leave the time tolerance at the default value of 10 . In the event that NTP fails, the 3-DNS uses the time_tolerance variable to maintain synchronization.	10
Timer Sync State	Specifies the interval (in seconds) at which the 3-DNS checks to see if it should change states (from Principal to Receiver or from Receiver to Principal).	30
Persist Cache	Specifies the interval (in seconds) at which the 3-DNS archives the paths and metrics data.	3600

Table 8.3 Time values and default settings

To configure global TTL and timer values using the Configuration utility

1. In the navigation pane, click **System**.
The System - General screen opens.

2. To configure the default TTL for wide IPs, type a new value in the **Default TTL** box.
3. To configure other TTL and timer values, click **Timers and Task Intervals** on the toolbar.
The System - Timers & Task Intervals screen opens.
4. Add the TTL and timer values settings.

For help on configuring the TTL and timer values settings, click **Help** on the toolbar.

To configure global TTL and timer values from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate or add the **globals** statement. The **globals** statement should be at the top of the file.
4. Use the syntax shown in Figure 8.12 to define global TTL and timer values.

```
globals {
  [ timer_get_3dns_data <number> ]
  [ timer_get_server_data <number> ]
  [ timer_get_host_data <number> ]
  [ timer_get_vs_data <number> ]
  [ timer_get_ecv_data <number> ]
  [ timer_get_path_data <number> ]
  [ timer_get_trace_data <number> ]
  [ timer_check_keep_alive <number> ]
  [ timer_check_pending_q_timeouts <number> ]
  [ timer_persist_cache <number> ]
  [ timer_sync_state <number> ]
  [ 3dns_ttl <number> ]
  [ server_ttl <number> ]
  [ host_ttl <number> ]
  [ vs_ttl <number> ]
  [ path_ttl <number> ]
  [ trace_ttl <number> ]
  [ default_ttl <number> ]
}
```

Figure 8.12 Syntax for configuring global TTL and timer values

Setting up load balancing for services that require multiple ports

Certain types of network traffic, such as FTP traffic or e-commerce traffic, require that more than one port be available in order for the client's requests to be properly handled. When you set up a load balancing configuration, you can define a port list for a wide IP. Before the 3-DNS selects a virtual server to receive a connection, it verifies that the virtual server is **up** and available to receive connection requests. When the 3-DNS receives a query, all of the ports in the port list must be available for each virtual server in the wide IP. If a virtual server does not have all ports in the port list available, the 3-DNS marks it as unavailable for load balancing.

To configure multiple ports for a wide IP using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
2. In the Wide IP column, click a wide IP name.
The Modify Wide IP screen opens.
3. On the toolbar, click **Port List**.
The Wide IP Port List screen opens.
4. Type a port number in the box or select a service from the list, then click the **Add** button.
5. Repeat step 4 for each port or service you need to add, then click **Update**.
The port list is added to the wide IP configuration.

To configure multiple ports for a wide IP from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement you want to edit.
4. Add the **port_list** line as indicated in bold in Figure 8.13.


```

wideip {
  address <ip_addr>
  port <port_number> | <"service name">
  name <"domain_name">
  [ alias <"alias_name"> ]
  [ ttl <number> ]
  [ port_list <port_number> <port_number> ... ]
  ...
  [ pool_lbmode <rr | ratio | ga | random | topology> ]
  [ pool definitions ...]
}

```

Figure 8.13 Enabling multiple ports with the *port_list* option

An example configuration using a port list

In this example, you are setting up a site for selling a product on the Internet. This site contains a non-secure area that contains the product catalog, and a secure area for placing orders. You can configure a wide IP so that clients are sent to a virtual server only when both the secure and non-secure ports are available.

The key entry for this configuration is **port_list**. The **port_list** entry specifies that requests can be sent to virtual servers in this pool only if ports 80 (non-secure) and 443 (secure) are available.

```

wideip {
  address          192.168.101.70
  port             80 // http
  port_list        80 443 // e-commerce
  name             "ssl.wip.domain.com"
  pool_lbmode      rr
  pool {
    name           "bigip_pool"
    ratio          2
    preferred      qos
    alternate      ratio
    address        192.168.101.70  ratio 7
    address        192.168.102.60  ratio 2
  }
  pool {
    name           "host_pool"
    ratio          1
    preferred      ratio
    address        192.168.104.50  ratio 2
    address        192.168.105.60  ratio 1
  }
}

```

Figure 8.14 Syntax for e-commerce services

For every virtual server address in the pool, a virtual server definition must exist for each port in the port list.

For the syntax example shown in Figure 8.14, the BIG-IP systems and hosts must have the following virtual servers defined:

```
192.168.101.70:80
192.168.101.70:443
192.168.102.60:80
192.168.102.60:443
192.168.104.50:80
192.168.104.50:443
192.168.105.60:80
192.168.105.60:443
```

Troubleshooting manual configuration problems

Adding a wide IP requires careful planning and use of correct syntax. We recommend using the Configuration utility to create wide IPs and pools so that the correct syntax is generated automatically in the **wideip.conf** file. However, we have included the following recommendations to make it easier for you to spot and resolve any configuration problems if you choose to create your configuration by editing the **wideip.conf** file.

- ◆ **Configuration utility**

The Configuration utility contains statistics screens that are useful in diagnosing problems, as they provide a snapshot of the 3-DNS network at any given time. To use the statistics screens, expand the **Statistics** item in the navigation pane, then click either **Wide IPs** or **Summary** (and scroll until you see the **Wide IP** table).

The Configuration utility also contains the Network Map, which allows you to see the relationships between your data centers, servers, and virtual servers, and the wide IPs and pools you created with the virtual servers. For information on working with the Network Map, click **Help** on the toolbar.

- ◆ **wideip.conf syntax**

If you configure wide IPs from the command line, use the **3dparse** utility to verify the **wideip.conf** syntax before you start **3dnsd**. To use the **3dparse** utility, type **3dparse** on the command line. For details on the **3dparse** utility, see the **3dparse** man page.

- ◆ **/var/log/messages**

If you encounter an error that you cannot trace, you can view the log file in the Configuration utility, or you can directly open the **/var/log/messages** file on your system. Using the UNIX **grep** utility, search for **3dnsd** (for example, **tail -100 /var/log/messages | grep 3dnsd**). This log file saves verbose error information, and should contain an explanation of the error.

- ◆ **BIND syntax**

If you are setting up the configuration from the command line, and you are running the 3-DNS in node mode, you may want to refer to one of the following BIND resources for help and background information:

- The O'Reilly & Associates book, *DNS and BIND*, Third Edition
- <http://www.isc.org/bind.html>



9

Network Map

- Introducing the Network Map
- Working with the Network Map

Introducing the Network Map

The 3-DNS Network Map is a dynamic map that illustrates the physical and logical objects in your network. With the Network Map, you can:

- Visualize the overall structure of your 3-DNS network
- Use the navigational tools to modify your network configuration
- View the enabled/disabled state of the various objects in your network

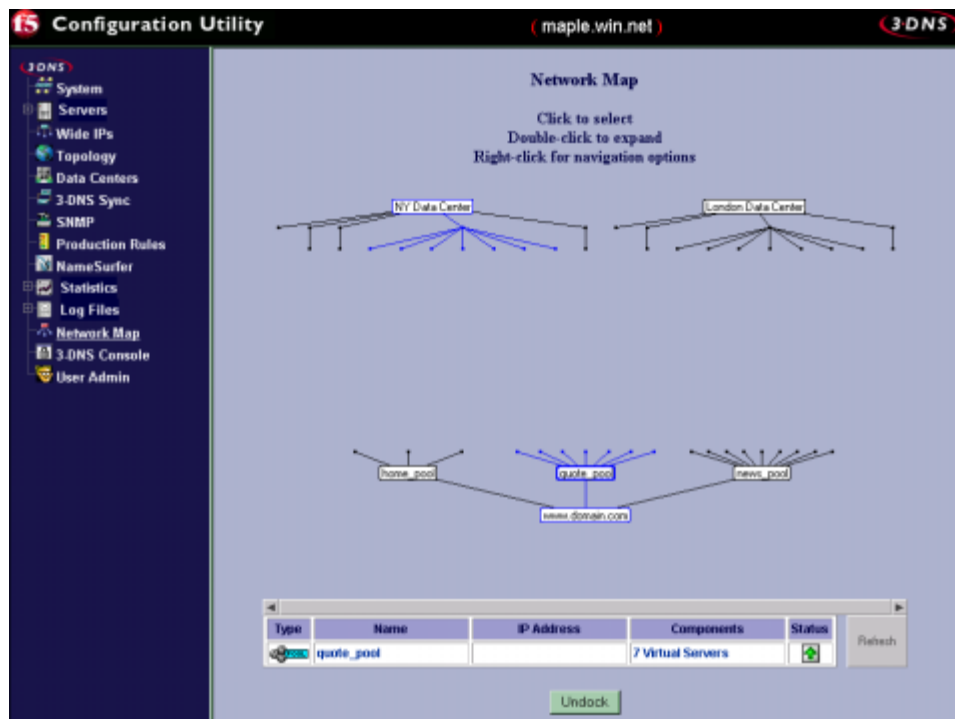


Figure 9.1 Example screen of the Network Map in the Configuration utility

In the Network Map, you can easily see how any component is related to the rest of the network, and how changes to the physical side of the network structure (for example, data centers or servers) can affect the logical side (for example, wide IPs or pools), and vice versa. As shown in Figure 9.1, the wide IP pool, **quote_pool**, is made up of virtual servers on a BIG-IP in the data center, **NY Data Center**.

Working with the Network Map

The Network Map is a highly interactive screen. Not only can you review and make changes to your 3-DNS configuration, but also you can use the information table to quickly check whether an object is enabled or disabled. The following sections describe some of the tasks you can do in the Network Map.

Viewing the Network Map

You can view the Network Map only from the Configuration utility.

To view the Network Map using the Configuration utility

1. In the navigation pane, click **Network Map**.
The Network Map screen opens.
2. Click **Undock** if you want to open a popup screen of the Network Map. For more information on working with the Network Map, click **Help** on the toolbar.

Using the Network Map to review and modify the network configuration

The Network Map contains the following objects: data centers, servers, wide IPs, pools, virtual servers. You can double-click any object on the Network Map to expand the object. The relationship of that object to the rest of the network becomes readily apparent, as the components of that object are highlighted in blue throughout the map. For example, if you double-click a data center, the data center expands, displaying and highlighting all of the servers that reside in that data center. Toward the bottom of the map, all wide IPs that contain a virtual server that belongs to the servers in the selected data center are also highlighted. You can continue to double-click the objects to narrow your scope.

From the Network Map, you can also navigate to the screens where you configure the various objects. You do this by right-clicking the object name. A popup menu opens, displaying various options from which you can choose, depending on what part of that object you want to configure. For example, if you right-click a wide IP name, and from the popup menu select **Configure**, the Modify Wide IP screen opens, where you can modify the settings for the wide IP definition.

Using the information table on the Network Map

When you double-click any object on the Network Map, the information table at the bottom of the Network Map screen displays the following details about that object:

- Object type
- Object name

- Object IP address
- Any child objects for the highlighted object
- Object status

You can also refresh the Network Map by clicking the Refresh button next to the information table.



10

Production Rules

- Controlling network traffic patterns with production rules
- Setting up production rules in the Configuration utility
- Working with the production rules scripting language

Controlling network traffic patterns with production rules

Production rules are a policy-based management tool that you can use to dynamically change how the 3-DNS distributes connections across the network. You can also use production rules to send system administrators notifications of specific events. Production rules are based on triggers, such as time of day, current traffic patterns, or current traffic volume. For example, you can configure a production rule that changes the load balancing mode to Quality of Service during your peak business hours, and you can configure a production rule that notifies you when the number of name resolution requests exceeds a specific number.

You can create production rules that apply to the system in general, or you can create production rules for specific wide IPs.

If you want to configure basic production rules, we recommend that you use the Configuration utility. If you want to create custom production rules, you should review the following section, *Working with the production rules scripting language*, on page 10-5, which describes the scripting language you use to configure production rules from the command line. You may also want to contact a technical support engineer for additional assistance with complex configurations.

Setting up production rules in the Configuration utility


The Configuration utility uses a wizard-style format to help you set up production rules. The screen prompts that you see during the configuration process vary, depending on the items you select in each screen. However, to configure any production rule, you perform three basic steps:

- ◆ **Define the type of rule**
The two types of production rules are: global production rules, and wide IP production rules.
- ◆ **Define the rule trigger**
The two types of rule triggers are: a set time or time interval, and a specific system event.
- ◆ **Define the action taken**
The two basic types of rule actions are: to send user-definable messages to log files or email accounts, and to change specific load balancing settings.

The following sections discuss each production rule option in detail, and provide all of the information you need to complete the production rule using the wizard.

Viewing, adding, and deleting production rules

When you click **Production Rules** in the Configuration utility, the Production Rules wizard screen opens. The screen displays the list of existing global and wide IP production rules. You can add a new rule by clicking the **Add Production Rule** toolbar button, which starts the production rule wizard. The wizard prompts you to specify the various production rule options, and then allows you to review your selections before you save the production rule to the configuration.

Note that you can modify existing production rules by clicking the rule name in the list, and you can delete a production rule at any time by clicking the Delete button  next to the rule name.

Choosing the rule type

The first step in the production rule wizard is to choose whether the production rule is a global production rule or a wide IP production rule.

- ◆ **Global production rules**

Global production rules send messages to log files or to specific email accounts, based on a set time interval or on standard events. The standard events are listed and described in Table 10.2, on page 10-8.

- ◆ **Wide IP production rules**

Wide IP production rules are based either on the time of day, or on standard events. Wide IP production rules can change the current load balancing modes for the preferred, alternate, or fallback methods; they can reconfigure ratio settings for individual virtual servers; and they can reconfigure the coefficients for Quality of Service mode. Wide IP production rules can also send messages to log files or email accounts.

After you choose a rule type, the wizard prompts you to name the rule and allows you to add a brief description of the rule.

Defining time-based triggers

The next step in the wizard prompts you to choose a trigger for the production rule. You can set up two basic types of triggers: time-based triggers and event-based triggers. This section describes the options for the time-based triggers, and the following section describes options for the event-based triggers. Once you review the information for the type of trigger you want to set up, go to *Choosing the action taken*, on page 10-4.

Time-based triggers include two types: global production rules trigger on set time intervals, while wide IP production rules trigger at specific times on specific days. To set a time interval for a global production rule, you define the number of seconds that elapse between each action the production rule executes.

A wide IP production rule can trigger at a specific time of day, on a specific day of the week, on a specific date, or at a specific time on a specific date. The following procedures explain how to set up each type of time trigger, in the wizard, for wide IP production rules.

To apply a time of day variable

1. From the Time Variable table, select **Time**.
2. In the **Start Time** box, specify the hour and minute you want the production rule action to begin.
3. In the **Stop Time** box, select the hour and minute you want the production rule action to stop.

Once you define the time of day that triggers the production rule, you continue with the wizard and begin to define the production rule action.

To apply a day of the week variable

1. From the Time Variable table, select **Day**.
A table opens from which you select the day to start and stop the action.
2. From the **Start Day** box, select the day you want the production rule action to begin.
3. From the **Stop Day** box, select the day you want the production rule action to stop.

Once you define the day of the week that triggers the production rule, you continue with the wizard and begin to define the production rule action.

To apply a date variable

1. From the Time Variable table, select **Date**.
A table opens from which you select the date to start and stop the action.
2. In the **Start Date** box, type the date you want the production rule action to begin (mm/dd/yyyy).
3. In the **Stop Date** box, type the date you want the production rule action to stop (mm/dd/yyyy).

Once you define the date that triggers the production rule, you continue with the wizard and begin to define the production rule action.

To apply a combined date and time variable

1. From the Time Variable table, select **Date/Time**.
Two tables open and you select the start and stop dates and times.

2. In the **Start Date** box, type the date you want the production rule action to begin (mm/dd/yyyy).
3. In the **Stop Date** box, type the date you want the production rule action to stop (mm/dd/yyyy).
4. In the **Start Time** box, specify the hour and minute you want the production rule action to begin.
5. In the **Stop Time** box, select the hour and minute you want the production rule action to stop.

Once you define the date and time that triggers the production rule, you continue with the wizard and begin to define the production rule action.

Defining event-based triggers

Both global production rules and wide IP production rules can be triggered by standard events, such as when a name resolution process begins. Wide IP production rules support two additional types of event-based triggers. You can set a wide IP production rule to trigger when a specific LDNS server makes a name resolution request, or to trigger when a user-specified number of name resolution requests are received by the 3-DNS.

The standard events that can trigger both global and wide IP production rules are described in Table 10.2, on page 10-8.

Choosing the action taken

After you specify the production rule trigger, the wizard prompts you to choose the action that the production rule takes. Note that the actions that a production rule can take depend in part on whether the production rule is a global rule or a wide IP rule. For example, both global production rules and wide IP production rules can send user-defined messages to log files, or to specific email accounts, but only wide IP production rules can alter load balancing modes. The actions that you can choose for a production rule are:

- ◆ **Sending user-defined messages**
Both global and wide IP production rules can send user-defined messages to the **syslog** file, or to a specific email account.
- ◆ **Changing the load balancing mode settings**
Wide IP production rules can change load balancing mode settings for the wide IP. You can change the preferred, alternate, and fallback methods, and you can change QOS coefficient settings.
- ◆ **Changing virtual server ratios**
You can change virtual server ratios to alter the distribution load when the load balancing mode is set to Ratio.
- ◆ **Specifying a virtual server to return**
You can specify that the 3-DNS returns a specific virtual server, rather than choosing a virtual server using load balancing.

Once you specify an action, the production rules wizard prompts you to review all of the production rule settings, and then saves the production rule to the configuration.

Working with the production rules scripting language

The production rules scripting language uses constructs and statements that are similar in syntax to Perl script and the C programming language. If you have a good working knowledge of Perl or C, you may want to create your own custom production rules. You can use the guidelines in this section in conjunction with the examples provided both here and in the sample **wideip.conf** file (installed on the 3-DNS).

If you need to add custom production rules to your configuration, but you do not want to work out the implementation yourself, you can contact your vendor for assistance.

Inserting production rules in the wideip.conf file

Production rules are part of the **wideip.conf** file, and you can either insert them directly in the file, or you can store them in a separate file and include them by reference. If you want to use the Configuration utility to manage the 3-DNS configuration, you must store production rules configured from the command line in a separate file, and include them by reference. If you attempt to use custom production rules in a file that you edit using the Configuration utility, the production rule syntax may become corrupt.

WARNING

*If you include custom production rules directly in the **wideip.conf** file, you must edit and maintain the **wideip.conf** file from the command line; you cannot use the Configuration utility for configuration administration.*

Executing and managing production rules from the command line

The language that you use to specify production rules is **3dscrip**. Production rules must have the following attributes in **3dscrip**:

- Each production rule is uniquely identified by a label.
- Each production rule can be deleted using its label.
- All production rules at the global scope can be deleted.
- All production rules at the wide IP-pool scope can be deleted.
- Each production rule can be replaced.
- Each production rule can be annotated with a character string.

The **3dscrip** language manages and executes production rules according the following guidelines:

- The **3dscrip**t language supports conditional execution of production rules using the **if** statement. You can use **if** statements in wide IP production rules, and in global production rules only if they are embedded within a **when** or an **every** statement.
- The **3dscrip**t language supports event-driven execution of production rules using the **when** statement. You can use the **when** statement only in global production rules.
- The **3dscrip**t language supports periodic execution of production rules using the **every** statement. You can use the **every** statement only in global production rules.

The following sections describe how to work with the components of the **3dscrip**t language.

Working with the if statement

```
if(conditional-expression) { <action> ... } [ else { <action> ... } ]
```

The **if** statement is a standard statement that defines an event condition that triggers a production rule action. Typically you use **if** statements in wide IP production rules. An **if** statement must adhere to the following guidelines:

- The **if** statement can be specified in the scope of a wide IP **pool** statement.
- The **if** statement can be nested in another **if** statement.
- Multiple **if** statements can be specified in the same scope.
- The nesting of **if** statements is limited only by the memory capacity of the 3-DNS.

- The precedence of logical, relational, and unary operators is the same as in ANSI-c.

If statement parameters and operators	Can contain or be one of these:
conditional-expression	A primitive-expression A primitive-expression followed by a relational-operator, followed by a primitive-expression A primitive-expression followed by an arithmetic-operator, followed by a primitive-expression Two conditional-expressions joined by a logical-operator
primitive-expression	A keyword which is evaluated when the conditional-expression is evaluated An intrinsic function which is evaluated when the conditional-expression is evaluated A literal value enclosed in full quotes A conditional-expression enclosed in parentheses A unary-operator followed by a conditional-expression enclosed in parentheses
logical-operators	Logical OR () Logical AND (&&)
relational-operators	Equality (==) Not equal (!=) Greater than (>) Greater than or equal to (>=) Less than (<) Less than or equal to (<=)
arithmetic-operator	modulus (mod)
unary operators	Unary negation (!) Unary minus (-)
keywords	day, time, date, datetime, ldns_ip, wip_ip, wip_name, wip_num_resolves, preferred, alternate, fallback, rtt, completion_rate, hops, packet_rate, topology
intrinsic functions	isLdnsInNet (Ip address, mask) isLdnsInAS (IP address, mask)

Table 10.1 Components of the *if* statement

Working with the when statement

```
when(event) { <action> ... }
```

The **when** statement is a standard statement that defines a specific event condition that triggers a production rule action. A **when** statement can be used only in global production rules, and it must adhere to the following guidelines:

- The **when** statement can be specified at the top scope of the **wideip.conf** file, after the **wide IP** definition(s) and before the **topology** statement.
- Multiple **when** statements can be specified in the same scope.
- Nesting of **when** statements is not allowed.

The production rule event triggers are described in Table 10.2.

Event triggers	Description
ResolveNameBegin	The production rule takes action each time the 3-DNS receives a new resolution request.
ResolveNameEnd	The production rule takes action each time the 3-DNS completes a name resolution.
FallbackToStatic	The production rule takes action each time the fallback load balancing method is used in a wide IP.
SIGINT	The production rule takes action each time the 3-DNS receives a SIGINT command.
SIGHUP	The production rule takes action each time the 3-DNS receives a SIGHUP command.
ReapPaths	The production rule takes action each time the 3-DNS reaps obsolete path information.
CRC_Failure	The production rule takes action each time iQuery communication on the 3-DNS experiences a CRC failure.
DownServer	The production rule takes action each time the 3-DNS detects that another 3-DNS, BIG-IP, or host server becomes unavailable.
DownVS	The production rule takes action each time the 3-DNS detects that a virtual server becomes unavailable.
DoneINT	The production rule takes action after the wideip.conf file is read on startup (a one-time event).
DoneConfigFile	The production rule takes action each time the 3-DNS configuration is re-read (for example, when a 3ndc reload command is issued).

Table 10.2 Standard production rule event triggers

Working with the every statement

```
every(<seconds>) { <action> ... }
```

The **every** statement is a standard statement that defines a time interval at which the production rule action triggers, such as every 60 seconds. An **every** statement can be used only for a global production rule, and it must adhere to the following guidelines:

- The **every** statement can be specified at the top scope of the **wideip.conf** file, after the wide IP definition(s) and before the **topology** statement.
- Multiple **every** statements can be specified in the same scope.
- Nesting of **every** statements is not allowed.

Defining production rule actions

The production rules language supports the following actions. Not all actions apply to all production rule types. For example, the actions that change load balancing settings are valid only for wide IP production rules. Actions such as defining a log string can be used in either global production rules or wide IP production rules. Each action below specifies which production rule types can use it.

Production rule actions	Description	Production rule type
preferred <lbmode>	This action changes the preferred load balancing method in a wide IP.	Wide IP production rule only
alternate <lbmode>	This action changes the alternate load balancing method in a wide IP.	Wide IP production rule only
fallback <lbmode>	This action changes the fallback load balancing method in a wide IP.	Wide IP production rule only
log (<string>)	This action sends the specified string to the syslog utility, which writes the string to the syslog file.	Wide IP production rule Global production rule
log2mail (<string>)	This action sends the specified string to the Sendmail utility, which creates a mail message and forwards it to the administrative email account specified for Sendmail (see the log2mail man page for details about log2mail syntax).	Wide IP production rule Global production rule

Table 10.3 Descriptions of production rule actions

Production rule actions	Description	Production rule type
vs(<ip>:<port>).ratio <n>	This action changes the ratio setting for a specific virtual server in a wide IP pool.	Wide IP production rule only
return_vs(<ip:port>)	This action skips the load balancing process and instead returns the specified virtual server to the requesting client.	Wide IP production rule only

Table 10.3 Descriptions of production rule actions

Production rule examples

There are a variety of custom production rules that you may want to implement or expand on for your own network. Following are examples of these three custom production rules:

- Load balancing according to time of day
- Load balancing according to local DNS server
- Hacker detection

Using production rules to load balance according to time of day

You can set up production rules ahead of time to deal with future needs and client demands for events. For example, say your company has a software distribution scheduled for release next Tuesday at 5:00 p.m. Pacific Standard Time. The new software will be available for download from the FTP sites at that time, and you expect that during the first week, traffic will be 10 times what it normally is, with frequent bursts during standard work hours, 7 a.m. to 6 p.m. However, the client base spans four time zones with an FTP server farm on the east coast in New York (**192.168.101.50**), and another on the west coast in Los Angeles (**192.168.102.50**). The 3-DNS is located on the east coast and runs on Eastern Standard Time. You are willing to accept some network latency in return for guaranteed connections.

Figure 10.1 shows a sample production rule that handles the connections according to the anticipated load at specific times of the day.

```
wideip {
  address 192.168.101.50:21
  name "ftp.domain.com"
  pool {
    preferred ratio
    address 192.168.101.50 ratio 2
    address 192.168.102.50 ratio 1
    rule "ftp_balance"
    // Night time: qos
    if(time > "21:00" && time < "07:00") {
      preferred leastconn
    }
    else {
      preferred ratio
      // East Coast
      rule "east" if(time < "10:00") {
        vs.(192.168.101.50).ratio 3
        vs.(192.168.102.50).ratio 1
      }
      // Both coasts are at peak demand
      else {
        rule "both" if(time < "18:00") {
          vs.(192.168.101.50).ratio 1
          vs.(192.168.102.50).ratio 1
        }
        // West Coast
        else {
          vs.(192.168.101.50).ratio 1
          vs.(192.168.102.50).ratio 3
        }
      }
    }
  }
}
```

Figure 10.1 Load balancing by time of day

Using production rules to load balance according to LDNS

One interesting application of production rules is that you can create a rule that is activated when a specific local DNS server makes a name resolution request. The following example is based on a web site published in three languages: English, Spanish, and Japanese. Suppose that the addresses in the network **10.10.0.0** are allocated to Japanese speakers, and the addresses

in the network **10.11.0.0** are allocated to Spanish speakers. The production rule shown in Figure 10.2 uses the address of the requesting LDNS to determine which virtual server should receive the connection.

```
wideip {
  address 192.168.101.50:80
  name "www.domain.com"
  pool {
    rule "Japanese" if(isLdnsInNet(10.10.0.0, 255.255.0.0)) {
      return_vs(192.168.103.50:80)
    }
    else {
      rule "Spanish" if(isLdnsInNet(10.11.0.0, 255.255.0.0)) {
        return_vs(192.168.102.50:80)
      }
      else { // assume English
        return_vs(192.168.101.50:80)
      }
    }
  }

  address 192.168.101.50 // English
  address 192.168.102.50 // Spanish
  address 192.168.103.50 // Japanese
}
}
```

Figure 10.2 Load balancing by IP address of LDNS

Using production rules for hacker detection

Another interesting example of triggering a production rule based on the requesting LDNS server is to take evasive action against known hackers attempting to access your system. The production rule shown in Figure 10.3 sends the hacker to a special server, rather than flat out rejecting the connection. As an alternative, you can change the rule to return a non-routable or non-existent address.

```
when(ResolveNameBegin) {
  rule "roach_motel" if(isLdnsInNet(10.20.30.4, 255.255.255.0)) {
    // Send this guy to our "roach motel" for hackers.
    // This address doesn't need to be listed in any wideip pool.
    // This address is reserved for us to watch hackers under the microscope.
    log2mail("Hacker $ldns_ip came back")
    return_vs(192.168.1.46:80)
  }
}
```

Figure 10.3 Sending a hacker to a specific server

A related example, shown in Figure 10.4, illustrates a production rule that deals with attacks against iQuery communications. The production rule would warn you if the 3-DNS detected a hack attempt against the iQuery protocol, based on a communication failure.

```
Rule "iQuery_hacked" when(CRC_Failure) {  
    log2mail("Got CRC Failure")  
}
```

Figure 10.4 *Detecting an iQuery failure due to potential attack*



||

Scripts

Working with scripts

3-DNS ships with several scripts to simplify many configuration and maintenance tasks. This chapter provides information about the functionality of these scripts. If you plan on performing a task from the command line that uses a script, you should find this section helpful. Many scripts correspond to commands on the 3-DNS Maintenance menu, so you may want to also review Chapter 2, *3-DNS Maintenance Menu*.

◆ Note

Before you edit a script, make a backup copy of the original.

3dns_add script

The **3dns_add** script allows you to add a new 3-DNS to an existing sync group in your network. The **3dns_add** script copies all configuration information from an existing 3-DNS onto the new system. For more details on using this script, refer to the *3-DNS Administrator Guide*, Chapter 5, *Adding a 3-DNS to an Existing Network*.

3dns_admin_start script

The **3dns_admin_start** script corresponds to the **Restart 3-DNS Configuration Utility** command on the 3-DNS Maintenance menu. This command restarts the 3-DNS web server, which hosts the Configuration utility.

3dns_dump script

The **3dns_dump** script saves the current state of the **3dnscd** cache to a new **/var/3dns/etc/wideip.conf** file.

3dns_web_config script

The **3dns_web_config** script corresponds to the **Reconfigure 3-DNS Configuration Utility** command on the 3-DNS Maintenance menu. This script lets you make configuration changes to the 3-DNS web server, which hosts the Configuration utility.

3dns_web_passwd script

The **3dns_web_passwd** script corresponds to the **Change/Add Users for 3-DNS Configuration Utility** command on the 3-DNS Maintenance menu. This script secures the 3-DNS web server using basic authentication. This script lets you provide restricted or administrative access to the 3-DNS web

server for selected users only, and assigns passwords for those users. Users with restricted access have access to the statistics area only. Users with administrative access have access to all areas of the 3-DNS web server.

3dnsmaint script

The **3dnsmaint** script opens the 3-DNS Maintenance menu. See Chapter 2, *3-DNS Maintenance Menu*, for more information.

3dprint script

The **3dprint** script corresponds to the **Dump 3dnssd Statistics** command on the 3-DNS Maintenance Menu. This script lets you view these statistics screens on the command line:

- ◆ **3-DNS**
Displays statistics about each 3-DNS in your network; the statistics include such things as whether the system is enabled or disabled, the number of packets per second traveling in and out of the 3-DNS during the last sample period, and the name of the sync group to which each 3-DNS belongs.
- ◆ **BIG-IP**
Displays statistics about all BIG-IP systems known to the 3-DNS; the statistics include such things as the number of virtual servers each BIG-IP manages, and the number of times the 3-DNS resolves requests to those virtual servers.
- ◆ **Hosts**
Displays statistics about all hosts known to the 3-DNS; the statistics include such things as the number of times that the 3-DNS resolves requests to the host, and the number of virtual servers that the hosts manage.
- ◆ **Virtual Servers**
Displays statistics about all BIG-IP, EDGE-FX, and host virtual servers; the statistics include such things as the server state, and the number of times it has received resolution requests.
- ◆ **Paths**
Displays path statistics, such as round trip time, packet completion rate, the remaining time-to-live (TTL) before a path's metric data needs to be refreshed.
- ◆ **Local DNS**
Displays statistics collected for local DNS servers; the statistics include such things as the number of resolution requests received from a given LDNS, and the protocol that the 3-DNS is using to probe the LDNS.
- ◆ **Wide IPs**
Displays statistics about each wide IP defined on the 3-DNS; the statistics include such things as load balancing information, and the remaining TTL before the wide IP's metrics data needs to be refreshed.

- ◆ **Globals**
Displays statistics about the **globals** sub-statements; the statistics include such things as the current and default values for each of the **globals** sub-statements, and whether you have to restart **3dnscd** when you make changes to the parameters.
- ◆ **Summary**
Displays summary statistics, such as the 3-DNS version, the total number of resolved requests, and the load balancing methods used to resolve requests.
- ◆ **Data Centers**
Displays statistics about the data centers, and their servers, in your network. The statistics include such things as the names of the data centers, the name or IP address of the servers in the data center, and whether the data center is enabled or disabled.
- ◆ **Sync Groups**
Displays statistics about each sync group in your network. The statistics include such things as the name of the sync group, whether **3dnscd** is running on each 3-DNS, whether the **big3d** agent is running on each 3-DNS, the name and IP address of the 3-DNS, and whether the 3-DNS is a principal or receiver.

3ndc script

The **3ndc** script starts the **3ndc** utility, which is described in the **3ndc** man page.

big3d_install script

The **big3d_install** script corresponds to the **Install and Start big3d** command on the 3-DNS Maintenance menu. This script installs and starts the appropriate version of the **big3d** agent on each BIG-IP, EDGE-FX Cache, and GLOBAL-SITE that the 3-DNS knows about. This script is useful for 3-DNS updates.

The **big3d_install** script performs the following procedure on each BIG-IP, EDGE-FX Cache, or GLOBAL-SITE:

1. Stops the running **big3d** agent process.
2. Uses a matrix file to determine which version of the **big3d** agent to copy to the BIG-IP, EDGE-FX Cache, or GLOBAL-SITE. The matrix file lists the version numbers for all BIG-IP systems, EDGE-FX Caches, and GLOBAL-SITE systems known to the 3-DNS, and the version numbers of the **big3d** agent running on each BIG-IP, EDGE-FX Cache, and GLOBAL-SITE.
3. Adds the following to the end of the **/etc/rc.conf** file:

```
big3d_enabled="yes"
```
4. Starts **/usr/sbin/big3d**.

For configuration options, see the **big3d** man page.

big3d_restart script

The **big3d_restart** script corresponds to the **Restart big3d** command on the 3-DNS Maintenance menu. This script stops and restarts the **big3d** agent on each BIG-IP, EDGE-FX Cache, and GLOBAL-SITE known to the 3-DNS.

big3d_version script

The **big3d_version** script corresponds to the **Check remote versions of big3d** command on the 3-DNS Maintenance menu. This script displays the version numbers for all BIG-IP systems, EDGE-FX Caches, and GLOBAL-SITE systems known to the 3-DNS, as well as the version numbers of the **big3d** agent running on those systems.

config_ssh script

The **config_ssh** script corresponds to the **Configure SSH communication with remote devices** command on the 3-DNS Maintenance menu. All 3-DNS scripts and synchronization require secure communications between systems. Any time you add a new 3-DNS, BIG-IP, EDGE-FX Cache, or GLOBAL-SITE to a network, you can run the **config_ssh** script, and if no **ssh** key exists on the system, the script configures **ssh** access.

◆ **Note**

This script is not available on the non-crypto version of the 3-DNS.

edit_lock script

The **edit_lock** script lets you safely edit a specified file that is synchronized between 3-DNS systems in a sync group. This script creates a temporary version of the original file, and this temporary file replaces the original file when you are finished editing it. If you do not use this script to edit a file, there is the danger that a partial file might be synchronized to other 3-DNS systems in the sync group.

To use this script, type the following, at the command line:

```
edit_lock <file name>
```

edit_wideip script

The **edit_wideip** script corresponds to the **Edit 3-DNS Configuration** command on the 3-DNS Maintenance menu. This script opens the **wideip.conf** file for editing, copies it to all other 3-DNS systems in the 3-DNS sync group, and restarts the **3dnsd** utility.

install_key script

The **install_key** script corresponds to the **Generate and Copy iQuery Encryption Key** command on the 3-DNS Maintenance menu. This script starts the **F5makekey** program, and generates a seed key for encrypting communications between the 3-DNS systems and (if you have any in your network) BIG-IP systems, EDGE-FX Caches, or GLOBAL-SITE systems. The **install_key** script creates and distributes the iQuery key to all BIG-IP systems, EDGE-FX Caches, GLOBAL-SITE systems, and other 3-DNS systems in your network.

◆ Note

This script is not available on the non-crypto version of 3-DNS.

To start the **F5makekey** program, type the following at the command line, in the **/usr/local/bin** directory:

```
f5makekey
```

The seed value is located in **/etc/F5key.dat** and contains a random length (12-52) of random content (1-255), created by the **F5makekey** program. This array of values is used by MD-160, a one-way hash function, to generate a key (7 characters in length) for the Blowfish encryption algorithm.

syncd_checkpoint script

The **syncd_checkpoint** script creates a checkpoint file. A *checkpoint file* is a compressed tar file that contains an archive of the files that are synchronized.

You can run this script with or without arguments. If you run **syncd_checkpoint** without specifying arguments, the script creates the following default checkpoint file:

```
/var/tmp/staging/checkpoint/default.tar.gz
```

◆ Note

*All checkpoint file names have a **.tar.gz** suffix.*

The **syncd_checkpoint** script can take the following optional arguments:

```
syncd_checkpoint [-c <name>] [-i]
```

The options for `syncd_checkpoint` are defined in Table 11.1.

Option	Description
<code>-c <name></code>	Creates a checkpoint file with the specified file name. You can also specify a non-default path for the file, unless the path starts with a slash (<code>/</code>). The default path for checkpoint files is <code>/var/3dns/staging/checkpoint/</code> . The <code>syncd_checkpoint</code> script automatically appends a <code>.tar.gz</code> extension to the end of the file name.
<code>-i</code>	Runs the script in an interactive session, which means that you are prompted for a file name.

Table 11.1 Optional arguments for the `syncd_checkpoint` script

syncd_rollback script

The `syncd_rollback` script decompresses a checkpoint file, which contains an archive of all synchronized files. This has the effect of replacing the current files with the files archived in the checkpoint file.

The `syncd_rollback` script can take the following optional arguments:

```
syncd_rollback [-c] [-c <name>] [-r] [-u] [-i]
```

◆ Note

When you run this script from the command line, you must use the `-r`, `-u`, or `-i` option.

The options for `syncd_rollback` are defined in Table 11.2.

Option	Description
<code>-c</code>	Unrolls the most recently created checkpoint file, whether it is in the default location or elsewhere.
<code>-c <name></code>	Unrolls the specified checkpoint file, whether it is in the default location or elsewhere. It is not necessary to end the name with <code>.tar.gz</code> , as this suffix is assumed.
<code>-r</code>	Restores archived files with their old timestamps. This means that if any of the synchronized files were updated on a remote 3-DNS, the updated files will overwrite any older files contained in the checkpoint file.
<code>-u</code>	Restores archived files with updated timestamps with the current time. This means that the files in the checkpoint are synchronized to the remote 3-DNS systems and overwrite the existing files on the remote 3-DNS systems.
<code>-i</code>	Runs the script in an interactive session, which means that you are prompted for option information.

Table 11.2 Optional arguments for the `syncd_rollback` script

syncd_start script

The **syncd_start** script corresponds to the **Restart syncd** command on the 3-DNS Maintenance menu. This script restarts the **syncd** daemon if it is already running, or starts it if it is not. You can run this script with or without arguments. If you run **syncd_start** without specifying arguments, the script starts or restarts **syncd**. The **syncd_start** script can take the following optional arguments:

```
syncd_start [-c] [-c <name>] [-r] [-u] [-i]
```

◆ Note

*When you use the **-c** option, you must also use either the **-r** or **-u** option.*

The options for **syncd_start** are defined in Table 11.3.

Option	Description
-c	Before restarting syncd , unrolls the most recently created checkpoint file, whether it is in the default location or elsewhere.
-c <name>	Before restarting syncd , unrolls the specified checkpoint file, whether it is in the default location or elsewhere. It is not necessary to end the name with .tar.gz , as this suffix is assumed.
-r	Restores the archived files with their old timestamps. This means that if any of the synchronized files were updated on a remote 3-DNS, the updated files overwrite the rolled back files.
-u	Restores the archived files with updated timestamps to the current time. This means that the files in the checkpoint file overwrite any updated files on remote 3-DNS systems.
-i	Runs the script in an interactive session, which means that you are prompted for option information.

*Table 11.3 Optional arguments for the **syncd_restart** script*

syncd_stop script

The **syncd_stop** script corresponds to the **Stop syncd** command on the 3-DNS Maintenance menu. This script stops the **syncd** daemon if it is running. You can run this script with or without arguments. If you run **syncd_stop** without specifying arguments, the script simply stops **syncd**. The **syncd_stop** script can take the following optional arguments:

```
syncd_stop [-c] [-c <name>] [-i]
```

The options for **syncd_stop** are defined in Table 11.4.

Option	Description
-c	Creates a checkpoint file in the default location before stopping syncd .
-c <name>	Creates a checkpoint file with the specified name and path before stopping syncd .
-i	Runs the script in an interactive session, which means that you are prompted for option information.

Table 11.4 Optional arguments for the **syncd_stop** script



12

SNMP

- Working with SNMP on the 3-DNS
- Configuring SNMP on the 3-DNS
- Configuring options for the checktrap.pl script
- Configuring the 3-DNS SNMP agent using the Configuration utility
- Configuring host SNMP settings on the 3-DNS
- Configuring the SNMP agent on host servers

Working with SNMP on the 3-DNS

3-DNS ships with a customized simple network management protocol (SNMP) agent and management information base (MIB). This chapter describes the management and configuration tasks with which you can configure the 3-DNS SNMP agent.

The 3-DNS SNMP agent and 3-DNS MIB allow you to monitor the 3-DNS by configuring traps for the SNMP agent or by polling the system with a standard network management station. The 3-DNS SNMP agent has the following options to ensure secure management:

- Community names
- TCP wrappers
- View access control mechanism (VACM)

Using the Configuration utility, you can configure the 3-DNS SNMP agent to send traps to your network management system. You can also set up custom traps by editing several configuration files.

◆ WARNING

If you want to monitor the 3-DNS using the SEE-IT Network Manager, you must configure the SNMP agent on the 3-DNS.

Configuring SNMP on the 3-DNS

To use SNMP on the 3-DNS, you must complete the following tasks:

- ◆ Download the 3-DNS MIBs and load them into your network management station
- ◆ Modify the following configuration files:
 - **/etc/hosts.allow**
 - **/etc/snmpd.conf**
 - **/etc/3dns_snmptrap.conf**
 - **/etc/syslog.conf**
- ◆ Configure options for the **checktrap** script

◆ Note

If you are configuring the 3-DNS module on a BIG-IP, you configure any SNMP settings using the BIG-IP Configuration utility.

Downloading the MIBs

The 3-DNS includes a proprietary 3-DNS SNMP MIB. This MIB is specifically designed for use with the 3-DNS. You can configure the SNMP settings in the Configuration utility or on the command line.

SNMP management software requires that you use the MIB files associated with the device. You can obtain the following three MIB files from the 3-DNS directory `/usr/local/share/snmp/mibs`, or you can download the files from the **Additional Software Downloads** section of the Configuration utility home screen. The files you need are:

- ◆ **3dns.my**
This is a vendor MIB that contains specific information for properties associated with specific 3-DNS functionality, such as load balancing.
- ◆ **rfc1611.my**
This is a DNS server MIB (RFC 1611) that provides standard management information.
- ◆ **UCD-SNMP-MIB.txt**
This is a MIB-II (RFC 1213) that contains specific management information for the UC-Davis SNMP agent.

For information about the objects defined in **3dns.my**, refer to the descriptions in the object identifier (OID) section of the MIB file. For information about the objects defined in **rfc1611.my**, refer to RFC 1611.

Understanding configuration file requirements

You need to make changes to several configuration files on the 3-DNS before using the SNMP agent. Once you change these configuration files, you must restart the SNMP agent. The files are discussed in the following sections.

`/etc/hosts.deny`

The `/etc/hosts.deny` file must be present to deny, by default, all UDP connections to the SNMP agent. The contents of this file are as follows:

```
ALL : ALL
```

`/etc/hosts.allow`

The `/etc/hosts.allow` file specifies the hosts that are allowed to access the SNMP agent. You can configure access to the SNMP agent with the `/etc/hosts.allow` file in one of two ways:

- By typing in an IP address, or list of IP addresses, that are allowed to access the SNMP agent.
- By typing in a network address and mask to allow a range of addresses in a subnet to access the SNMP agent.

You can specify a list of addresses that you want to allow access to the SNMP agent. Addresses in the list must be separated by blank space or by commas. Use the following syntax:

```
daemon: <IP address> <IP address> <IP address>
```

In the following example, the SNMP agent accepts connections from the specified IP addresses only:

```
snmpd: 128.95.46.5 128.95.46.6 128.95.46.7
```

For a range of addresses, the basic syntax is as follows, where **daemon** is the name of the daemon, and **NETWORKADDRESS/MASK** specifies the network that is allowed access:

```
daemon: NETWORKADDRESS/MASK
```

For example, the following example sets the **snmpd** daemon to allow connections from the **128.95.46.0/255.255.255.0** address range:

```
snmpd: 128.95.46.0/255.255.255.0
```

The previous example allows the 256 possible hosts from the network address **128.95.46.0** to access the SNMP daemon. You may also use the keyword **ALL** to allow access for all hosts or all daemons.

◆ Note

*If you prefer, instead of modifying this file from the command line, you can use the Configuration utility to specify the hosts that are allowed to access the SNMP agent. See **To set SNMP properties using the Configuration utility**, on page 12-7.*

/etc/snmpd.conf

The **/etc/snmpd.conf** file controls most aspects of the SNMP agent. This file is used to set up and configure certain traps, passwords, and general SNMP variable names.

A few of the necessary variables are listed below:

◆ System Contact Name

The System Contact is a MIB-II simple string variable defined by almost all SNMP systems. It usually contains a user name and an email address. This is set by the **syscontact** key.

◆ Machine Location (string)

The Machine Location is a MIB-II variable that is supported by almost all systems. It is a simple string that defines the physical location of the system. This is set by the **syslocation** key.

◆ Community String

The community string clear text password is used for basic SNMP security. This also maps to VACM groups, but for initial read-only access, it is limited to only one group.

◆ **Trap Configuration**

Trap configuration is controlled by these entries in the `/etc/snmpd.conf` file:

- **trapsink <host>**
This sets the host to receive trap information. The `<host>` variable is an IP address.
- **trapport <port>**
This sets the port on which traps are sent. There must be one **trapport** line for each **trapsink** host.
- **trapcommunity <community string>**
This sets the community string (password) for sending traps. Once set, it also sends a trap upon startup: **coldStart(0)**.
- **authtrapenable <integer>**
Set this variable to **1** so that traps can be sent for authentication warnings. Set the variable to **2** to disable it.

*Note: To change the trap port, be sure the **trapport** line precedes the **trapsink** line. If you use more than one **trapport** line, there must be one **trapport** line before each **trapsink** line. The same is true for **trapcommunity**; if you use more than one **trapcommunity** line, there must be one **trapcommunity** line before each **trapsink** line.*

◆ **System IP Setting**

You must set the system IP address using the **sysip** command; if this setting is not present, the **checktrap.pl** script fails to send all 3-DNS-specific traps. Use the following syntax to set the system IP address:

```
sysip <3-DNS IP address>
```

◆ **Note**

*If you prefer, instead of modifying this file from the command line, you can use the Configuration utility to set these SNMP properties. See **To set SNMP properties using the Configuration utility**, on page 12-7.*

/etc/3dns_snmptrap.conf

The configuration in the **/etc/3dns_snmptrap.conf** file determines which messages generate traps and what those traps are. The file includes OIDs, traps, and regular expression mappings. The configuration file specifies whether to send a specific trap based on a regular expression. An excerpt of the configuration file is shown in Figure 12.1.

```
# Default traps.
.1.3.6.1.4.1.3375.1.2.2.2.0.1 (SNMP_TRAP: VS.*?state change green.*?red) VIRTUAL SERVER
GREEN TO RED

.1.3.6.1.4.1.3375.1.2.2.2.0.2 (SNMP_TRAP: VS.*?state change red.*?green) VIRTUAL SERVER
RED TO GREEN

.1.3.6.1.4.1.3375.1.2.2.2.0.3 (SNMP_TRAP: SERVER.*?state change green.*?red) SERVER
GREEN TO RED

.1.3.6.1.4.1.3375.1.2.2.2.0.4 (SNMP_TRAP: SERVER.*?state change red.*?green) SERVER RED
TO GREEN

.1.3.6.1.4.1.3375.1.2.2.2.0.5 (SNMP_TRAP: iQuery message from big3d) CRC FAILURE
```

Figure 12.1 Excerpt from the **/etc/3dns_snmptrap.conf** file

Some of the OIDs have been permanently mapped to specific 3-DNS events. The OIDs that are permanently mapped for the 3-DNS include:

- Virtual server green to red
- Virtual server red to green
- Server green to red
- Server red to green
- CRC failure
- Pool red to green
- Pool green to red
- 3-DNS active to standby
- 3-DNS standby to active

To see messages that are triggering an SNMP trap, look in the **var/3dns/log/3dns.log** file.

/etc/syslog.conf

To generate traps, you must configure **syslog** to send syslog lines to **checktrap.pl**. If the syslog lines match the specified regular expression in the **3dns_snmptrap.conf** file, the **checktrap.pl** script generates a valid SNMP trap. The following line in the **/etc/syslog.conf** file causes the **syslog** utility to send the specified log output to the **checktrap.pl** script. The **checktrap.pl** script then compares the logged information to the **3dns_snmptrap.conf** file to determine if a trap should be generated.

```
local2.warning | exec /sbin/checktrap.pl.
```

◆ Note

*If you uncomment this line, make sure you restart **syslogd**.*

Configuring options for the **checktrap.pl** script

The **checktrap.pl** script reads a set of lines from standard input. The script checks each line against a set of regular expressions. If a line matches a regular expression, the script sends an SNMP trap.

The following options are available for the **checktrap.pl** script.

◆ SNMP configuration file

This file contains the SNMP variables. The **checktrap.pl** script gets trap configuration information from this file. The default is **/etc/snmpd.conf**.

```
snmpd_conf_file=<snmp configuration file>
```

◆ SNMP trap configuration file

This file contains the regular expression to SNMP trap OID mappings. It also contains a description string that is added to the trap message. The default is **/etc/3dns_snmptrap.conf**.

```
trapd_conf_file=<snmp trap configuration file>
```

◆ SNMP trap program

This program sends the SNMP trap. This program should be the **snmptrap** program included with the 3-DNS. The default is **/sbin/snmptrap**.

```
trap_program=<snmp trap program>
```

◆ Date removal

This option turns off automatic date removal. Normally, each input line is expected to begin with a date. Typically, this date is removed before the trap is sent. This option keeps the date information in the trap. If you do not add this option, the date is removed from the trap by default.

```
no_date_strip
```

◆ Usage

This option prints a usage string.

```
usage
```

Configuring the 3-DNS SNMP agent using the Configuration utility

You can use the Configuration utility to configure the following aspects of the 3-DNS SNMP agent:

- ◆ **Client access**

You can define a network address and netmask for a workstation from which SNMP requests are acceptable.

- ◆ **System information**

You can name a system contact, a machine location, and a community string.

- ◆ **Trap configuration**

You can enter a trap sink and a trap community.

To set SNMP properties using the Configuration utility

The Configuration utility provides sample SNMP settings for your reference. To use the 3-DNS SNMP MIB, you must replace these sample settings with settings appropriate to your environment and your specific SNMP management software.

1. In the navigation pane, click **SNMP**.
The SNMP Configuration screen opens.
2. Add the SNMP settings. For help on configuring the SNMP settings, click **Help** on the toolbar.

◆ **Note**

If you are configuring the 3-DNS module on a BIG-IP, you configure the SNMP settings in the BIG-IP Configuration utility.

Configuring host SNMP settings on the 3-DNS

After defining a host server, you need to configure its SNMP settings if you want to use SNMP host probing. Remember that you must first set up at least one SNMP prober factory on any 3-DNS, BIG-IP, EDGE-FX Cache, or GLOBAL-SITE that runs the **big3d** agent.

The SNMP factory collect can collect some or all of the following information from a host:

- Memory utilization
- CPU utilization
- Disk space utilization
- Kilobytes/second
- Current connections
- Packet rate

The 3-DNS gathers metrics for BIG-IP systems, EDGE-FX Caches, and several host servers. Refer to Table 12.1 for information on the host server types and the specific metrics that can be collected for each host type. To see the current performance of any of these server metrics, review the Metrics statistics screen.

Server Type or Operating System	Metrics collected:						
	Kilobytes/Second	Packets/Second	CPU	Memory	Disk	Current Connections	Nodes Up
BIG-IP	X	X				X	X
EDGE-FX Cache	X	X				X	
Alteon Ace Director	X					X	X
BSD, UC Davis	X	X	X	X	X	X	
CacheFlow	X	X	X			X	
Cisco CSS series	X	X				X	X
Cisco LocalDirector	X	X				X	
Cisco LocalDirector	X	X				X	
Cisco SLB						X	X
Extreme	X	X				X	X
Foundry ServerIron	X	X				X	X
Linux, UC Davis	X	X		X	X	X	
Sun Solaris	X	X	X			X	
Windows 2000 Server	X	X	X			X	
Windows NT 4.0	X	X	X	X		X	

Table 12.1 Server types and the metrics collected by the 3-DNS

◆ **Note**

The Cisco LocalDirector metric shows new connections per second rather than current connections.

To configure host SNMP settings using the Configuration utility

1. In the navigation pane, expand the **Servers** item, and click **Host**.

2. From the Host column, click a host server.
The Modify Host screen opens.
3. On the toolbar, click **SNMP Configuration**.
The Host SNMP Configuration screen opens.
4. Add the SNMP settings for the host. For help on configuring the SNMP settings for a host, click **Help** on the toolbar.

To configure host SNMP settings from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate or add the host **server** statement. (All **server** statements should appear after the **globals** statement and before **wideip** statements.)
4. Define the server type, address, name, prober, probe protocol, and port information as usual.
5. Add the **snmp** statement.
6. Define the virtual server information as usual.

Figure 12.2 shows the SNMP syntax for a host server in bold.

```
server {
  type host
  address <IP address>
  name <"host_name">
  probe_protocol <dns_dot | dns_rev | tcp | icmp>
  [ prober <IP address> ]
  port <port number> | service <"service name">
  [ snmp {
    agent <generic | ucd | solstice | ntserv | win2kserve | ciscold | ciscold2 | ciscold3
    | foundry | arrowpoint | alteon | cacheflow>
    port <port number>
    community <"community string">
    timeout <seconds>
    retries <number>
    version <SNMP version>
  } ]
  vs {
    address <virtual server IP address>
    port <port number> | service <"service name">
    [ probe_protocol <dns_dot | dns_rev | tcp | icmp> ]
  }
}
```

Figure 12.2 Configuring host SNMP settings

Configuring the SNMP agent on host servers

For host probing to work properly, you need to verify that the SNMP agent is properly configured on the host itself. We recommend that you refer to the documentation provided with your host SNMP software for complete configuration information.



13

Topology

- Working with Topology load balancing
- Setting up topology records
- Using the Topology load balancing mode in a wide IP
- Using the Topology load balancing mode in a pool
- Understanding user-defined regions
- Working with the topology statement in the wideip.conf file

Working with Topology load balancing

To use the Topology load balancing mode, you first set up topology records in a topology statement. Once you have defined a topology statement, you can set up Topology load balancing among pools in a wide IP, or within a pool. Note that if you do not create a topology statement, and you configure Topology as the load balancing mode, the 3-DNS load balances requests using the Random mode.

The crypto 3-DNS includes a database that maps IP addresses to geographic locations. With this database, the system can use the geographic attributes of local DNS servers (LDNS) to direct traffic.

The following sections describe how to create a topology statement, and how to set up Topology load balancing.

◆ Note

*Topology is also a coefficient in the QOS equation. If you have configured a Topology statement, the topology coefficient is calculated in the QOS score. For more information on the QOS equation and the QOS score, see the **3-DNS Administrator Guide**, Chapter 9, Working with Quality of Service.*

Setting up topology records

A **topology record** has three elements: an LDNS location endpoint, a virtual server location endpoint, and a relative weight. The location endpoints can be one of the following:

- A IP subnet (CIDR definition)
- A wide IP pool (managed by the 3-DNS)
- A data center (managed by the 3-DNS)
- A country (based on the ISO 3166 top-level domain codes, as specified by IANA, the Internet Assigned Numbers Authority)
- A continent
- An Internet service provider (ISP) (for LDNS location endpoints only)
- A user-defined region

The relative weight, or **topology score**, for the topology record allows the 3-DNS to evaluate the best resolution option for a DNS request. The not (!) operator, when used in a topology record, indicates location endpoints not equal to the specified value.

A **topology statement** is composed of one or more topology records. Figure 13.1 is an example of a topology statement, with two topology records, as it appears in the Configuration utility.

Add Topology Records	
Server	LDNS
IP Subnet	IP Subnet
<input type="checkbox"/> Not Equal	<input type="checkbox"/> Not Equal
Weight	
Add	

Current Topology Records	
pool.origin	continent.North America 1000
pool.cache_farm	!continent.North America 1000

Remove

Figure 13.1 Example of a topology statement in the Configuration utility

Here is an explanation of how to interpret the topology statement in the preceding example. A wide IP pool labeled **origin** manages the virtual servers that are returned for DNS resolution requests sent by local DNS servers located in North America. A wide IP pool labeled **cache_farm** manages the virtual servers that are returned for DNS resolution requests sent by local DNS servers located anywhere except North America. When the 3-DNS receives a DNS resolution request from an LDNS located in North America, it evaluates the first topology record and assigns a score of 100, because the LDNS criteria matches. The system then evaluates the next topology record, and assigns a score of **0** because the LDNS criteria does not match. The system then routes the DNS request to the wide IP pool **origin** for resolution, because that topology record has the highest score.

To add topology records using the Configuration utility

1. In the navigation pane, click **Topology**.
The Manage Topology Records screen opens.
2. Add the settings for the topology records, and click **Add**. For assistance with the settings on this screen, click **Help** on the toolbar.

To remove topology records using the Configuration utility

1. In the navigation pane, click **Topology**.
The Manage Topology Records screen opens.
2. Select the topology record that you want to remove from the Current Topology Records list, and click **Remove**.
The topology record is removed from the topology statement. For assistance with the settings on this screen, click **Help** on the toolbar.

Using the Topology load balancing mode in a wide IP

You can use the Topology load balancing mode to distribute traffic among wide IP pools. You must have at least two pools configured in the wide IP. You can use the Topology load balancing mode with pools to direct traffic to a specific data center in your network, to a third-party network, or to a content delivery network.

To set up topology to distribute traffic among wide IP pools using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
The Wide IP List screen opens.
2. In the Wide IP column, click a wide IP name.
The Modify Wide IP screen opens.
3. In the **Pool LB Mode** box, select **Topology** as the load balancing mode for the wide IP.
4. Click **Update**.

To set up topology to distribute traffic among wide IP pools from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement that you want to edit.
4. Define **topology** as the **pool_lbmode** load balancing mode for the wide IP.

Figure 13.2 shows a sample wide IP definition where Topology is the load balancing mode among the pools in this wide IP configuration.

```
wideip {
  address 192.168.44.1
  name "www.domain.com"
  port 80
  pool_lbmode topology

  pool {
    name "cache_farm"
    fallback null
    address 192.168.44.10
    address 192.168.44.20
  }

  pool {
    name "origin"
    address 172.168.11.10
    address 172.168.11.20
  }
}
```

Figure 13.2 Example syntax for Topology pool load balancing in a wide IP

Using the Topology load balancing mode in a pool

In addition to setting up the Topology load balancing mode to select a pool within a wide IP, you can also set up the Topology load balancing mode to select a virtual server within a pool. However, you must configure the topology records before the 3-DNS can use the Topology load balancing mode within a pool. If you have no topology records in the topology statement, **Topology** does not appear, in the Configuration utility, as an option for the **Preferred**, **Alternate**, or **Fallback** load balancing modes for pools.

To set up topology load balancing within a pool using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
The Wide IP List screen opens.
2. In the Pools column, click a pool.
The Modify Wide IP Pools screen opens.
3. In the Pool Name column, click a pool name.
The Modify Load Balancing for [pool name] screen opens.
4. In the **Preferred** box, select **Topology** as the load balancing mode for the pool.
5. Click **Update**.
The change is added to the configuration.

To set up topology load balancing in a pool from the command line

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement you want to edit.
4. Define **topology** as the **preferred** (or **alternate** or **fallback**) load balancing mode for the pool.

The example in Figure 13.3 shows a sample wide IP definition where **topology** is the load balancing mode within a pool.

```
wideip {
  address 192.168.103.60
  port 80
  name "ntp.wip.domain.com"
  pool {
    name "poolA"
    preferred topology
    alternate rtt
    address 192.168.101.60 // New York
    address 192.168.102.60 // Los Angeles
    address 192.168.103.60 // Tokyo
  }
}
```

Figure 13.3 Example of Topology load balancing mode within a pool

Understanding user-defined regions

To further refine the topology load balancing capabilities of the 3-DNS, you can create custom topology regions. By adding user-defined regions to the topology statement, the 3-DNS can route traffic (client requests) to the best data center or wide IP based on the characteristics of your specific network.

You create a custom region by adding one or more region member types to the region member list. The region member types are: **Continent**, **Country**, **Data Center**, **IP Subnet**, **ISP**, **User-Defined Region**, and **Wide IP Pool**. Once you select a region member type, you then fill in the details about that region member and add it to the region member list. The region member options change based on the region member type that you select. When you have finished adding region members to your new region, the new region becomes an option on the Manage Topology screen, in the **Server** and **LDNS** boxes.

To create a user-defined region using the Configuration utility

1. In the navigation pane, click **Topology**.
The Manage Topology screen opens.
2. On the toolbar, click **Manage User-Defined Regions**.
The Region List screen opens.
3. On the toolbar; click **Add Region**.
The User-Defined Region screen opens.
4. On the User-Defined Region screen, add the settings you want for your custom region. For information on the specific settings, click **Help** on the toolbar.

To create a user-defined region from the command line.

1. Type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. Using the arrow keys, select **Edit 3-DNS Configuration**.
3. Using your preferred text editor (**vi** or **pico**), add the user-defined region. Use the syntax shown in Figure 13.4.

```
rdn user {  
  region {  
    name "US_no_AOL_or_STUFF"  
    cont."North America"  
    ! country."MX"  
    ! country."CA"  
    ! ISP."AOL"  
  }  
}
```

Figure 13.4 Syntax for user-defined regions

Working with the topology statement in the wideip.conf file

The **topology** statement, in the **wideip.conf** file, can include the following variables to specify pools, data centers, continents, countries, and user-defined regions, in addition to the traditional CIDR blocks, for both servers and local DNS servers.

Variable	Description
pool.<"pool_name">	Specifies a wide-IP pool for load balancing. Note that pool names can be duplicated across wide IPs. The name must be in quotation marks. Use this for server in a topology record.
datacenter.<"datacenter_name">	Specifies a data center for load balancing. The name must be in quotation marks. Use this for server in a topology record.
continent.<"continent_name">	Specifies one of the continents for load balancing: " North America ", " South America ", " Europe ", " Asia ", " Australia ", " Africa ", or " Antarctica ". The name must be in quotation marks. Use this for ldns in a topology record.
country.<"2-letter_code">	Specifies a country for load balancing using one of the two-letter country codes found in the file <code>/var/3dns/include/net.ccdb</code> . The name must be in quotation marks. Use this for ldns in a topology record.
isp."AOL"	For local DNS servers only, specifies the Internet service provider, America Online (AOL). The name must be in quotation marks.
user.<"region_name">	Specifies a user-defined region. The name must be in quotation marks.
!	The not (!) operator negates the meaning of an element in a topology record.
score	Specifies the relative weight, or score, for the topology record, which allows the 3-DNS to evaluate the best resolution option for a DNS request.

Table 13.1 Variables used in the *topology* statement

To add a **topology** statement to the include file `/var/3dns/include/topology.inc`, follow the format of this example.

```

topology {
// server      ldns      score
pool."origin"  cont."North America"  100
pool."cache_farm" !cont."North America"  100
datacenter"dc_1" user."Europe"  300
pool."origin"  user."headquarters"  200
}

```

Figure 13.5 Example of a *topology* statement

◆ **Note**

Use the **not** (!) notation in a topology statement to negate the meaning of an element, as shown in Figure 13.5.



A


3-DNS Configuration File

- Overview of the 3-DNS configuration file
- Working with statements
- Working with comments
- Understanding current values

Overview of the 3-DNS configuration file

The 3-DNS configuration file describes a network's data centers, servers (3-DNS systems, BIG-IP systems, EDGE-FX Caches, GLOBAL-SITE systems, and hosts), virtual servers, and the wide IPs and pools used for load balancing. The 3-DNS configuration file is called **wideip.conf**. Note that when you use the browser-based Configuration utility, all components of the 3-DNS configuration file are automatically generated and parsed.

◆ Note

*If you use the Configuration utility to configure the 3-DNS, and you want to see the **wideip.conf** configuration for a specific component, click the Configuration View button  when you see it in the Configuration utility.*

The **wideip.conf** file consists of two types of information: statements and comments. The **wideip.conf** file should include at least the following definitions.

- A **datacenter** statement
- At least one **server** statement defining a 3-DNS
- At least one virtual server, which is defined as part of a BIG-IP, EDGE-FX Cache, or host **server** statement
- A **wideip** statement, for load balancing

If the **wideip.conf** file lacks complete definitions, one of the following happens:

- If the file cannot be parsed, **3dnsd** does not start.
- If the file can be parsed, the 3-DNS reverts to standard DNS behavior.

To open the 3-DNS configuration file

1. At the command line, type **3dnsmaint**.
The 3-DNS Maintenance menu opens.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration**.

◆ WARNING

*We do not recommend opening the **wideip.conf** file in a text editor. Instead, use the **Edit 3-DNS Configuration** command on the 3-DNS Maintenance menu. This command allows you to edit and save the configuration file. This command also parses the configuration file and alerts you to any syntax errors.*

Using include files

Include files are files that contain configuration information about one aspect of your network, and are listed in the main configuration file (**wideip.conf**). For example, you can have one **include** file that defines the servers in your network, and another **include** file that defines all the wide IPs that are used for load balancing. Both files are listed in the **wideip.conf** file in place of the actual **server** and **wideip** statements.

Using **include** files reduces the size of the **wideip.conf** file and makes it easier to manage your configuration. 3-DNS automatically creates and implements **include** files whenever you configure your network setup using the Configuration utility.

◆ **Note**

*When the **wideip.conf** file is generated by the Configuration utility, any comments you incorporated from the command line are deleted.*

Syntax for include files

Use the following syntax when incorporating **include** files into a **wideip.conf** file.

```
include root_in "/config/3dns/include"
include root_out "/config/3dns/include"
include global <"file_name.inc">
include server <"file_name.inc">
include bigip <"file_name.inc">
include host <"file_name.inc">
include 3dns <"file_name.inc">
include datacenter <"file_name.inc">
include sync_group <"file_name.inc">
include wideip <"file_name.inc">
include 3dscrip <"file_name.inc">
include topology <"file_name.inc">
include geoloc <"netIana.inc">
include ldns <"ldns.inc">
include region <"file_name.inc">
include manifest <"file_name.inc">
```

Figure A.1 Syntax for include files

Definitions of include statements

Table A.1 lists the include statements, their descriptions, and their default file names.

Parameter	Description	Default
include root_in	Specifies the root directory from which to get include files. Enclose all file names in quotation marks.	"/config/3dns/include"
include root_out	Specifies the root directory in which to dump include files.	"/config/3dns/include"
include global	Specifies the name of the file that contains the globals statement.	"globals.inc"
include server	Specifies the name of the file that contains server statements.	"servers.inc"
include bigip	Specifies the name of the file that contains BIG-IP server statements.	"bigip.inc"
include host	Specifies the name of the file that contains host server statements.	"host.inc"
include 3dns	Specifies the name of the file that contains 3-DNS server statements.	"3dns.inc"
include datacenter	Specifies the name of the file that contains datacenter statements.	"datacenters.inc"
include sync_group	Specifies the name of the file that contains sync_group statements.	"sync_groups.inc"
include wideip	Specifies the name of the file that contains wideip statements.	"wideip.inc"
include 3dscrip	Specifies the name of the file that contains production rule configuration.	"prodrules.inc"
include topology	Specifies the name of the file that contains the topology statement.	"topology.inc"
include geoloc	Specifies the name of the file that contains the IP geo-classification database. It is important that you do not edit this statement.	"netlana.inc"
include ldns	Specifies the name of the file that contains information about local DNS servers and path information.	"ldns.inc"
include region	Specifies the name of the file that contains any region definitions statements.	"region.inc"
include manifest	Specifies the name of the file that the Configuration utility uses to manage any production rules generated by the utility. It is important that you do not edit this statement.	"/config/3dns/ .prodruledb/manifest"

Table A.1 Include file descriptions

Working with statements

A top-level 3-DNS statement begins with a keyword, and may be followed either by a value or by a block of sub-statements enclosed in braces ({}).

The 3-DNS platform supports the following top-level statements.

- ◆ **include**
The **include** statement lists any **include** files that are configured on the 3-DNS.
- ◆ **globals**
The **globals** statement defines system-level settings for any 3-DNS configuration options and sets the defaults for other statements.
- ◆ **server**
The **server** statement defines a 3-DNS, a BIG-IP and its virtual servers, an EDGE-FX Cache and its virtual servers, a GLOBAL-SITE, or a host machine and its virtual servers (if applicable).
- ◆ **datacenter**
The **datacenter** statement defines the group of 3-DNS, BIG-IP, EDGE-FX Cache, GLOBAL-SITE, and host systems that reside in a single physical location.
- ◆ **sync_group**
The **sync_group** statement defines the group of 3-DNS systems that synchronize their configuration settings and metrics data.
- ◆ **wideip**
The **wideip** statement defines a wide IP and its pools. A *wide IP* maps a domain name to a load balancing mode and a set of virtual servers (on a BIG-IP, EDGE-FX Cache, or host, or any combination thereof).
- ◆ **topology**
The **topology** statement contains the topology records that facilitate the topology load balancing mode (on its own and as part of the Quality of Service mode). Note that the **topology** statement is the preferred location for topology configuration information.

Syntax rules

Keep the following rules in mind when creating and editing statements in the **wideip.conf** file.

- ◆ **Statement order**
Statements should appear in this order in the **wideip.conf** file:
 - **globals**
 - most **include** statements (except the **include ldns** statement)
 - **server**
 - **datacenter**
 - **sync_group**
 - **wideip**
 - **include ldns**
- ◆ **Address and port specification**
For virtual servers (on a BIG-IP, an EDGE-FX Cache, or a host), the port specification must immediately follow the IP address specification. For

the port specification, you can use either a port number, or a service name. For example, you can use "**http**" instead of **80** to represent the HTTP protocol. The address and port specification can take any of the following forms:

```
address <ip_addr>:<port>
address <ip_addr>
port <port>
address <ip_addr>
service <"http">
```

- ◆ **Current values**

You may notice several current values (indicated by **cur_ "value"**) in the **wideip.conf** file; do not edit them unless you are instructed to do so by your vendor's technical support. For more information, see *Understanding current values*, on page A-43.

Typography in syntax examples

Certain characters are used to indicate whether a parameter is mandatory or optional, or whether you can use one parameter or another.

- ◆ **Mandatory parameters**

Angle brackets (< >) enclose mandatory parameters where you must type the data associated with a command.

- ◆ **Optional parameters**

These brackets ([]) enclose optional parameters.

- ◆ **Brackets**

These brackets ({ }) include the options available in a statement or sub-statement.

- ◆ **Choice of parameters**

A vertical bar (|) between two values means that either value is acceptable.

The globals statement

The **globals** statement sets up global options to be used by the 3-DNS, and must appear before any other statements in the **wideip.conf** file. Each **globals** sub-statement has a default setting, and you do not need to edit the **globals** statement unless you want to change a default setting. If the 3-DNS does not find a **globals** statement in the configuration file, the 3-DNS uses a **globals** block, with each option set to its default.

If you use a **globals** sub-statement more than once, the 3-DNS uses the last listed value and does not generate an error message. For example, if your **globals** statement contains the lines shown in the following figure, the 3-DNS uses the value **50**.

```
globals {  
  bigip_ttl 100  
  bigip_ttl 50  
}
```

*Figure A.2 Multiple **globals** sub-statements*

Syntax for the `globals` statement

The **globals** statement supports the following sub-statements. When you define a **globals** statement, you need only include those sub-statements that you want to change from the default.

```
globals {
  [ time_tolerance <number> ]
  [ encryption < yes | no > ]
  [ encryption_key_file <string> ]
  [ check_static_depends < yes | no > ]
  [ check_dynamic_depends < yes | no > ]
  [ default_persist_ttl < <number> s | m | h | d | w | m | y > ]
  [ default_probe_limit <number> ]
  [ persist_ldns < yes | no > ]
  [ persist_mask <ip address> ]
  [ drain_requests < yes | no > ]
  [ timer_get_3dns_data <number> ]
  [ timer_get_server_data <number> ]
  [ timer_get_host_data <number> ]
  [ timer_get_vs_data <number> ]
  [ timer_get_ecv_data <number> ]
  [ timer_get_path_data <number> ]
  [ timer_get_trace_data <number> ]
  [ timer_check_keep_alive <number> ]
  [ timer_persist_cache <number> ]
  [ timer_sync_state <number> ]
  [ dc_prefix <string> ]
  [ dns_ttl <number> ]
  [ 3dns_ttl <number> ]
  [ bigip_ttl <number> ]
  [ edgefx_ttl <number> ]
  [ host_ttl <number> ]
  [ vs_ttl <number> ]
  [ path_ttl <number> ]
  [ trace_ttl <number> ]
  [ default_ttl <number> ]
  [ rtt_timeout <number> ]
  [ rtt_sample_count <number> ]
  [ rtt_packet_length <number> ]
  [ rx_buf_size <number> ]
  [ tx_buf_size <number> ]
  [ dump_region < yes | no > ]
  [ dump_topology < yes | no > ]
}
```

Figure A.3 Syntax for the `globals` statement

```

[ qos_coeff_rtt <number> ]
[ qos_coeff_completion_rate <number> ]
[ qos_coeff_packet_rate <number> ]
[ qos_coeff_topology <number> ]
[ qos_coeff_hops <number> ]
[ qos_coeff_vs_capacity <number> ]
[ qos_coeff_kbps <number> ]
[ qos_factor_rtt <number> ]
[ qos_factor_completion_rate <number> ]
[ qos_factor_packet_rate <number> ]
[ qos_factor_topology <number> ]
[ qos_factor_hops <number> ]
[ qos_factor_vs_capacity <number> ]
[ qos_factor_kbps <number> ]
[ default_alternate < ga | null | random | ratio | static_persist |
packet_rate | leastconn | return_to_dns | rr | topology | vs_capacity
| kbps > ]
[ default_fallback < completion_rate | ga | hops | leastconn |
null | packet_rate | qos | random | ratio | return_to_dns |
rr | rtt | topology | vs_capacity | static_persist | kbps > ]
[ fb_respect_depends < yes | no > ]
[ fb_respect_acl < yes | no > ]
[ aol_aware < yes | no > ]
[ path_duration <number> ]
[ ldns_duration <number> ]
[ prober <ip_addr> ]
[ resolver_tx_buf_size <number> ]
[ resolver_rx_buf_size <number> ]
[ use_alternate_iq_port < yes | no > ]
[ multiplex_iq < yes | no > ]
[ paths_never_die < yes | no > ]
[ rtt_allow_probe < yes | no > ]
[ rtt_allow_hops < yes | no > ]
[ rtt_allow_frag < yes | no > ]
[ rtt_probe_protocol < dns_rev | dns_dot | udp | tcp | icmp > ]
[ datasize_system <number> ]
[ datasize_reap_pct <number> ]
[ default_iquery_protocol < udp | tcp > ]
[ traceroute_port <number> ]
[ do_dynamic < yes | no > ]
}

```

Figure A.3 Syntax for the **globals** statement

Figure A.4 shows an example of a valid **globals** statement.

```

globals {
  prober 192.168.101.2      // Default prober is New York 3-DNS
  encryption yes          // Encrypt iQuery
  path_ttl 2400            // Extend the life of path metrics
}

```

Figure A.4 Example syntax for the **globals** statement

Definition of globals sub-statements

The **globals** sub-statements and their parameters are described in the following sections.

Synchronization

The synchronization sub-statement specifies how the current 3-DNS handles synchronizing its database with the other 3-DNS systems in the network.

Parameter	Description	Default
time_tolerance	Specifies the variation of time allowed (in seconds) when comparing time stamps on files. The syncd daemon allows for slight variation in time stamps when it compares files during the synchronization process. If the difference between the two time stamps falls within the time_tolerance setting, the daemon considers the files to be the same and does not overwrite one with the other.	10

Table A.2 Synchronization sub-statement

Encryption

The encryption sub-statements specify whether the communication between the 3-DNS and a BIG-IP is encrypted.

Parameter	Description	Default
encryption	Specifies whether to enable encryption for iQuery events.	no
encryption_key_file	Specifies the location and name of the iQuery encryption key file.	"/etc/F5key.dat"

Table A.3 Encryption sub-statements

Dependencies

The dependencies sub-statements specifies whether the 3-DNS checks the availability of virtual servers or paths before the system sends a connection to a virtual server.

Parameter	Description	Default
check_static_depends	Specifies whether to check the availability of virtual servers on BIG-IP, EDGE-FX Cache, and host systems. Change this option to no if you want to test your configuration. Setting this option to no forces the virtual servers to have green (up) status indicators on the Virtual Server Statistics screen in the Configuration utility.	yes
check_dynamic_depends	Specifies that the 3-DNS checks the availability of a path before it uses the path for load balancing. Changing this option to no overrides the path_ttl and whether the last probe attempt was successful.	yes

Table A.4 Dependencies sub-statement

LDNS persistence

Dynamic load balancing modes depend on path information to resolve requests. The value for **persist_ldns** must be set to **yes** (the default) so that the 3-DNS stores and uses path information. If you use only static load balancing modes, you can set **persist_ldns** to **no** to conserve memory.

Parameter	Description	Default
persist_ldns	Specifies whether the 3-DNS records in its cache the IP addresses of all LDNS machines that make resolution requests.	yes

Table A.5 LDNS persistence sub-statement

Load balancing persistence

The load balancing persistence sub-statements define how the 3-DNS load balances persistent connections.

Parameter	Description	Default
default_persist_ttl	Specifies the length of time the 3-DNS retains persistent connections information before the information is purged.	3600
persist_mask	Specifies the significant bits of an LDNS IP address to use with the static_persist load balancing mode.	0xFFFFFFFF
drain_requests	Specifies whether load-balanced persistent connections are allowed to remain connected, until the TTL expires, when you disable a pool. When set to no , the connections are terminated immediately when the pool is disabled. This variable affects the persist setting in the load balancing sub-statement. See Table A.28, on page A-34, for more information.	yes

Table A.6 Load balancing persistence sub-statements

Periodic task intervals

The periodic task interval sub-statements define the frequency at which the 3-DNS refreshes the metrics information it collects.

Parameter	Description	Default
timer_get_3dns_data	Specifies how often the 3-DNS retrieves performance data for other 3-DNS systems in the sync group. You can enter a value between 1 and 4294967295 seconds.	20
timer_get_server_data	Specifies how often the 3-DNS refreshes 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE information. You can enter a value between 1 and 4294967295 seconds.	20

Table A.7 Periodic task interval sub-statements

Parameter	Description	Default
timer_get_host_data	Specifies how often the 3-DNS refreshes other host machine information. You can enter a value between 1 and 4294967295 seconds.	90
timer_get_vs_data	Specifies how often the 3-DNS refreshes virtual server information. You can enter a value between 1 and 4294967295 seconds.	30
timer_get_path_data	Specifies how often the 3-DNS refreshes path information (for example, round trip time or ping packet completion rate). You can enter a value between 1 and 4294967295 seconds.	120
timer_get_ecv_data	Specifies how often the 3-DNS refreshes ECV information. You can enter a value between 5 and 4294967295 seconds.	90
timer_get_trace_data	Specifies how often the 3-DNS retrieves traceroute data (the traceroute utility collects information on router hops between each data center and each LDNS). You can enter a value between 1 and 4294967295 seconds.	60
timer_check_keep_alive	Specifies how often the 3-DNS queries remote 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems. This value determines how often 3dnsd sends hello packets to each big3d agent in its configuration. You can enter a value between 1 and 4294967295 seconds.	60
timer_persist_cache	Specifies how often the 3-DNS writes the wideip.conf file from memory. You can enter a value between 1 and 4294967295 seconds.	300

Table A.7 Periodic task interval sub-statements

Data time-outs

The data time-out sub-statements set the amount of time for which metrics information is considered valid. After a time-out is reached, the 3-DNS refreshes the information.

Parameter	Description	Default
default_ttl	Specifies the default number of seconds that the 3-DNS considers the wide IP A record to be valid. If you do not specify a wide IP TTL value when you define a wide IP pool, the wide IP definition uses the default_ttl value.	30
3dns_ttl	Specifies the number of seconds that the 3-DNS considers performance data for the other 3-DNS systems to be valid.	60
bigip_ttl	Specifies the number of seconds that the 3-DNS can use BIG-IP information for name resolution and load balancing. You can enter a value between 1 and 4294967295 . The following relationship should be maintained: bigip_ttl is greater than timer_get_server_data . A 2:1 ratio is the optimal setting for this relationship.	60

Table A.8 Data time-outs sub-statements

Parameter	Description	Default
edgefx_ttl	Specifies the number of seconds that the 3-DNS can use EDGE-FX Cache information for name resolution and load balancing. You can enter a value between 1 and 4294967295 . The following relationship should be maintained: edge_ttl is greater than timer_get_server_data . A 2:1 ratio is the optimal setting for this relationship.	60
host_ttl	Specifies the number of seconds that the 3-DNS can use host information for name resolution and load balancing. You can enter a value between 1 and 4294967295 . The following relationship should be maintained: host_ttl is greater than timer_get_host_data .	240
vs_ttl	Specifies the number of seconds that the 3-DNS can use virtual server information (data acquired from a BIG-IP, EDGE-FX Cache, or host about a virtual server) for name resolution and load balancing. You can enter a value between 1 and 4294967295 . The following relationship should be maintained: vs_ttl is greater than timer_get_vs_data .	120
path_ttl	Specifies the number of seconds that the 3-DNS can use path information for name resolution and load balancing. You can enter a value between 1 and 4294967295 . The following relationship should be maintained: path_ttl is greater than timer_get_vs_data .	2400
trace_ttl	Specifies the amount of time (in seconds) that the 3-DNS considers traceroute data to be valid. You can enter a value between 1 and 4294967295 .	604800 (seven days)

Table A.8 Data time-outs sub-statements

Metrics collection

The metrics collection sub-statements define how the 3-DNS collects path information.

Parameter	Description	Default
rtt_timeout	Specifies how long the big3d agent waits for a probe. You can enter a value between 1 and 4294967295 seconds.	5
rtt_sample_count	Specifies the number of packets to send from the big3d agent to the LDNS to determine the path information between those two systems. You can type a value between 1 and 25.	3
rtt_packet_length	Specifies the length of packets, in bytes, to send from the big3d agent to the LDNS to determine the path information between those two machines. You can type a value between 64 and 500; the default value for this setting is 64.	64
rtt_probe_protocol	Determines which protocols the 3-DNS uses to probe LDNS servers to calculate RTT times, and in what order the protocols are used. You can specify the icmp , udp , tcp , dns_dot , and dns_rev protocols.	icmp

Table A.9 Metrics collection sub-statements

Resource limits

The resource limits sub-statements define the amount of memory on the 3-DNS that is allocated to sending and receiving metrics information.

Parameter	Description	Default
rx_buf_size	Specifies the maximum amount of socket buffer data memory the 3-DNS can use when receiving iQuery data. You can enter a value between 8192 and 65536 .	49152
tx_buf_size	Specifies the maximum amount of socket buffer data memory the 3-DNS can use when transmitting iQuery data. You can enter a value between 8192 and 65536 .	49152

Table A.10 Resource limits sub-statements

QOS values

The Quality of Service (QOS) load balancing mode distributes connections based on a path evaluation score. Using the QOS equation shown in Figure A.5, the Quality of Service mode compares paths between the LDNS and each virtual server included in the **wideip** statement. When you specify the Quality of Service load balancing mode, the 3-DNS load balances each new connection to the virtual server associated with the best (highest) path score.

```
score_path =
[(qos_coeff_packet_rate) * (1 / score_packet_rate)] +
(qos_coeff_rtt) * (1 / score_rtt)] +
[(qos_coeff_completion_rate) * (score_completion_rate)] +
[(qos_coeff_topology) * (score_topology)] +
[(qos_coeff_hops) * (score_hops)] +
[(qos_coeff_vs_capacity) * (score_vs_capacity)] +
[(qos_coeff_kbps) * (score_kbps)]
```

Figure A.5 QOS equation

The coefficients for the QOS score computation are defined in the **globals** statement, but you can override them within a **wideip** statement.

Parameter	Description	Default
qos_coeff_rtt	Specifies the relative weighting for round trip time when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	50
qos_coeff_completion_rate	Specifies the relative weighting for ping packet completion rate when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	5

Table A.11 QOS values sub-statements

Parameter	Description	Default
qos_coeff_packet_rate	Specifies the relative weighting for BIG-IP packet rate when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	1
qos_coeff_topology	Specifies the relative weighting for topology when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	0
qos_coeff_hops	Specifies the relative weighting for hops when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	0
qos_coeff_vs_capacity	Specifies the relative weighting for virtual server capacity when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	0
qos_coeff_kbps	Specifies the relative weighting for kilobytes per second when the load balancing mode is set to Quality of Service. You can enter a value between 0 and 4294967295 .	3
qos_factor_rtt	Specifies the factor used to normalize raw round trip time values when computing the QOS score.	10000
qos_factor_completion_rate	Specifies the factor used to normalize raw completion rate values when computing the QOS score.	10000
qos_factor_packet_rate	Specifies the factor used to normalize raw packet rate values when computing the QOS score.	10000
qos_factor_topology	Specifies the factor used to normalize raw topology values when computing the QOS score.	10
qos_factor_hops	Specifies the factor used to normalize raw hops values when computing the QOS score.	25
qos_factor_vs_capacity	Specifies the factor used to normalize raw virtual server capacity values when computing the QOS score.	1
qos_factor_kbps	Specifies the factor used to normalize raw kilobytes per second values when computing the QOS score.	1

Table A.11 QOS values sub-statements

Load balancing

The load balancing sub-statement defines the alternate and fallback load balancing modes.

Parameter	Description	Default
default_alternate	Defines the default alternate load balancing mode used when the preferred load balancing mode does not provide a resolution. You can override this setting in the wideip statement.	rr
default_fallback	Defines the default fallback load balancing mode used when the preferred and alternate load balancing modes do not provide a resolution. You can override this setting in the wideip statement.	return_to_dns
fb_respect_depends	Determines whether the 3-DNS respects virtual server status when load balancing switches to the specified fallback mode.	no
fb_respect_acl	Determines whether the 3-DNS imposes topology access control when load balancing switches to the specified fallback mode.	no
aol_aware	Determines whether the 3-DNS recognizes local DNS servers that belong to the Internet service provider, America Online (AOL).	yes

Table A.12 Load balancing sub-statements

Prober

The prober sub-statement defines the IP address of the machine that pings a host system to verify whether it is available. Typically, you use the IP address of the 3-DNS itself, but you can use other network servers.

Parameter	Description	Default
prober	Specifies the default prober for collecting host status, and is usually the 3-DNS IP address. Using this sub-statement is not necessary if the 3-DNS does not manage host virtual servers. When this option is set to 0 , the system's IP address is the implied value. This sub-statement can be overridden within the server statement.	0.0.0.0

Table A.13 Prober sub-statement

Buffer size

The buffer size sub-statements specify the maximum amount of UDP data that the 3-DNS can receive for wide IP DNS messages.

Parameter	Description	Default
resolver_rx_buf_size	Specifies the wide IP receive buffer size. The value is overridden only if it is larger than the one first assigned by the kernel.	98304
resolver_tx_buf_size	Specifies the wide IP send buffer size.	24576

Table A.14 Buffer size sub-statements

Reaping

The 3-DNS stores local DNS server and network path data in memory. The amount of data that can be held in memory at any given time is based on the amount of memory in the 3-DNS. **Reaping** is the process of finding the least-used data in memory and deleting it.

The default reaping values are adequate for most configurations. Contact your technical support representative if you want to make changes to them.

Parameter	Description	Default
datasize_system	Specifies the amount of RAM that the 3-DNS reserves for system usage, such as non-3-DNS specific processes.	64MB
datasize_reap_pct	Specifies what percentage of memory that the 3-DNS frees up during the reap process.	15
path_duration	Specifies the number of seconds that a path remains cached after its last access. You can type a value between 60 and 2147483648).	604800 (7 days)
ldns_duration	Specifies the number of seconds that an inactive LDNS remains cached. Each time an LDNS makes a request, the clock starts again. You can type a value between 60 and 2147483648 .	2419200 (28 days)

Table A.15 Reaping sub-statements

iQuery port options

The iQuery port options determine which port (or ports) the 3-DNS uses to send and receive iQuery traffic.

Parameter	Description	Default
use_alternate_iq_port	Determines whether the 3-DNS runs iQuery traffic on port 245 (the port used in older configurations), or on port 4353 , the iQuery port registered with IANA. The default setting, yes , uses port 4353 . To use port 245 , change this setting to no . This setting is used only by UDP-based traffic.	yes
multiplex_iq	Determines whether the 3-DNS uses the ephemeral ports for iQuery traffic returned from the big3d agent. The default setting forces iQuery traffic to use a single port defined by use_alternate_iq_port for all incoming iQuery traffic.	yes

Table A.16 iQuery port options sub-statements

Probing

The 3-DNS uses probing to collect path metrics. The 3-DNS then uses the metrics to make traffic distribution and load balancing decisions.

Parameter	Description	Default
default_probe_limit	Specifies a limit on the number of times the 3-DNS probes a path. With the default setting, there is no limit on path probes.	0
paths_never_die	Specifies that dynamic load balancing modes can use path data even after the TTL for the path data has expired. We recommend that you change this setting to yes , which has the effect of requiring that the 3-DNS always uses path data even if the path's TTL expires.	no
check_dynamic_depends	Specifies that the 3-DNS checks the availability of a path before it uses the path for load balancing. Changing this option to no overrides the path_ttl and whether the last probe attempt was successful. This parameter does not prevent the refreshing of path metrics.	yes
rtt_allow_probes	Specifies that the 3-DNS issues probe requests for path metrics to local DNS servers. You can change this setting to no to turn off path probing.	yes
rtt_allow_hops	Specifies that the 3-DNS should collect hops metrics when probing paths.	yes
rtt_allow_frag	Specifies that the 3-DNS should break each probe packet into smaller packets when probing paths.	no
probe_protocol	Specifies the protocol that the 3-DNS uses to probe local DNS servers. The default is icmp . The other available protocols are: dns_rev , dns_dot , udp , and tcp .	icmp

Table A.17 Probing sub-statements

The server statement

The **server** statement defines the characteristics associated with a particular 3-DNS, BIG-IP, EDGE-FX Cache, GLOBAL-SITE, or host. A **server** statement contains the following information:

- The type of server: 3-DNS, BIG-IP, EDGE-FX Cache, GLOBAL-SITE, or host
- The IP address and host name of the server
- If the server is a BIG-IP, EDGE-FX Cache, or host, the set of virtual servers that is available on it
- Dynamically collected information about the server, its virtual servers and ports, and the paths between the server and LDNS

Because available sub-statements vary by server type, the syntax and examples for each type are listed separately. All sub-statements are defined in the table starting on page A-25.

Syntax for the server statement (3-DNS)

The following **server** statement syntax applies to 3-DNS systems only. Note that this **server** statement does not define virtual servers; the purpose of defining a 3-DNS is to set up the **big3d** agent to obtain path probing information.

```
server {
  type 3dns
  address <IP address>
  [ name <"3dns_name"> ]
  [ iquery_protocol [ udp | tcp ] ]
  [ remote {
    secure <yes | no>
    user <"user name">
  } ]
  [ interface {
    address <NIC IP address>
    address <NIC IP address>
  } ]
  [ factories {
    prober <number>
    snmp <number>
    hops <number>
    ecv <number>
  } ]
}
```

Figure A.6 Server statement syntax for defining a 3-DNS

Figure A.7 shows an example of the syntax to use in defining a 3-DNS.

```
// New York
server {
  type 3dns
  address 192.168.101.2
  name "3dns-newyork"
  iquery_protocol udp
  remote {
    secure no
    user "root"
  }
  factories {
    prober 5
    snmp 1
    ecv 5
  }
}
```

Figure A.7 Example syntax for defining a 3-DNS

Syntax for the server statement (BIG-IP)

The following **server** statement syntax applies to BIG-IP systems and their virtual servers only.

```
server {
    type bigip
    address <IP address>
    [ name <"bigip_name"> ]
    [ iquery_protocol [udp | tcp] ]
    [ remote {
        secure <yes | no>
        user <"user name">
    } ]
    [ interface {
        address <NIC IP address>
        address <NIC IP address>
    } ]
    [ prober <ip address> ]
    [ limit {
        [ kbytes_per_sec <number> ]
        [ pkts_per_sec <number> ]
        [ current_conns <number> ]
    } ]
    [ factories {
        prober <number>
        snmp <number>
        hops <number>
        ecv <number>
    } ]
    vs {
        address <virtual server IP address>
        port <port number> | service <"service name">
        [ ratio <number> ]
        [ limit {
            [ kbytes_per_sec <number> ]
            [ pkts_per_sec <number> ]
            [ current_conns <number> ]
        } ]
        [ depends_on {
            <IP address>:<port number> //example 10.10.10.10:443
        } ]
        [ ratio <number> ]
        [ translate {
            <IP address>:<port number>
        } ]
    }
}
```

Figure A.8 Server statement syntax for defining a BIG-IP

Figure A.9 shows an example of the syntax to use in defining a BIG-IP.

```
server {
    type          bigip
    address       192.168.101.40
    name          "bigip-newyork"
    iquery_protocol  udp
    remote {
        secure    yes
        user      "administrator"
    }
    # Tell 3-DNS about the 2 interfaces on a BIG-IP
    interface {
        address   192.168.101.41
        address   192.168.101.42
    }
    # Change the number of factories doing the work at big3d
    factories {
        prober    6
        snmp      1
        hops      2
        ecv       1
    }
    vs {
        address   192.168.101.50
        service   "http"
        translate {
            address 10.0.0.50
            port    80
        }
    }
    vs {
        address   192.168.101.50:25 // smtp
        translate {
            address 10.0.0.50:25
        }
    }
}
```

Figure A.9 Example syntax for defining a BIG-IP

Syntax for the server statement (EDGE-FX Cache)

This **server** statement syntax applies to EDGE-FX Caches only.

```
server {
  type edgefx
  address <IP address>
  [ name <"edgefx_name"> ]
  [ limit {
    [ kbytes_per_sec <number> ]
    [ pkts_per_sec <number> ]
    [ current_conns <number> ]
    [ cpu_avail <number> ]
    [ disk_avail <number> ]
    [ mem_avail <number> ]
  } ]
  [ iquery_protocol [ udp | tcp ] ]
  [ remote {
    secure <yes | no>
    user <"user name">
  } ]
  [ factories {
    prober <number>
    hops <number>
    snmp <1> { //required
      agent edgefx
      version 2
      community <"public">
    }
  } ]
  vs {
    address <virtual server IP address>
    port <port number> | service <"service name">
    [ ratio <number> ]
    [ limit {
      [ cpu_avail <number> ]
      [ disk_avail <number> ]
      [ mem_avail <number> ]
      [ kbytes_per_sec <number> ]
      [ pkts_per_sec <number> ]
      [ current_conns <number> ]
    } ]
  }
}
```

Figure A.10 Example syntax for defining an EDGE-FX Cache

Syntax for the server statement (GLOBAL-SITE)

The following **server** statement syntax applies to GLOBAL-SITE systems only.

```
server {
  type gsite
  address <IP address>
  [ name <"gsite_name"> ]
  [ iquery_protocol [ udp | tcp ] ]
  [ remote {
    secure <yes | no>
    user <"user name">
  } ]
  [ factories {
    prober <number>
    hops <number>
    snmp <number>
  } ]
}
```

Figure A.11 Example syntax for defining a GLOBAL-SITE

Syntax for the server statement (host)

The following **server** statement syntax applies to hosts only. Note that the **snmp** sub-statement is necessary only if you want the **big3d** agent to use an SNMP agent on the host to collect additional metrics information. For more information on configuring these settings, see Chapter 12, *SNMP*.

```
server {
    type host
    address <IP address>
    [ name <"host_name"> ]
    [ probe_protocol <tcp | icmp | dns_rev | dns_dot> ]
    [ prober <IP address> ]
    [ port <port number> | service <"service name"> ]
    [ snmp {
        agent <generic | ucd | solstice | ntserver | win2kserver |
        ciscold | ciscold2 | ciscold3 | foundry | arrowpoint | alteon
        | cacheflow>
        port <port number>
        community <"community string">
        timeout <seconds>
        retries <number>
        version <SNMP version>
    } ]
    [ limit {
        [ kbytes_per_sec <number> ]
        [ pkts_per_sec <number> ]
        [ current_conns <number> ]
        [ cpu_avail <number> ]
        [ disk_avail <number> ]
        [ mem_avail <number> ]
    } ]
    vs {
        address <virtual server IP address>
        port <port number> | service <"service name">
        [ probe_protocol <tcp | icmp | dns_rev | dns_dot> ]
    }
    [ ratio <number> ]
    [ limit {
        [ kbytes_per_sec <number> ]
        [ pkts_per_sec <number> ]
        [ current_conns <number> ]
        [ cpu_avail <number> ]
        [ disk_avail <number> ]
        [ mem_avail <number> ]
    } ]
    [ depends_on {
        <IP address>:<port number> //example 10.10.10.10:443
    } ]
}
```

Figure A.12 Server statement syntax for defining a host

Figure A.13 shows an example of the syntax to use in defining a host.

```
server {
    type          host
    address       192.168.104.40
    name          "host-tokyo"
    probe_protocol icmp
    snmp {
        agent      ucd
        community  "public"
        version    1
    }
    vs {
        address    192.168.104.50:25
        limit {
            kbytes_per_second 15000
        }
    }
    vs {
        address    192.168.104.50:80
        limit {
            kbytes_per_second 15000
        }
    }
}
```

Figure A.13 Example syntax for defining a host

Definition of server sub-statements

The **server** statement supports the following sub-statements. Note that available sub-statements vary by server type.

Address information

The address information sub-statements specify the name, address, and type of each server. Depending on the type of server you are configuring, you may need to specify a probe protocol, prober IP address, and port number.

Table A.18 lists the parameters of the address information sub-statement.

Parameter	Description
type	Indicates whether the specified server is a 3-DNS, BIG-IP, EDGE-FX Cache, GLOBAL-SITE, or host.
address	Specifies the IP address of the 3-DNS, BIG-IP, EDGE-FX Cache, GLOBAL-SITE, or host.
name	Specifies the name of the 3-DNS, BIG-IP, EDGE-FX Cache, GLOBAL-SITE, or host. You must enclose all names in quotation marks.
iQuery_protocol	Specifies the iQuery transport option, TCP or UDP.
probe_protocol	Specifies the protocol method to use for probing this host: icmp , tcp , dns_rev , or dns_dot . Applies to hosts only.

Table A.18 Address information sub-statements

Parameter	Description
prober	Specifies the IP address of the system probing the host. This IP address points to a BIG-IP, a 3-DNS, a GLOBAL-SITE, or an EDGE-FX Cache that runs the big3d agent. The big3d agent actually probes the host and virtual servers to verify whether the host or a particular virtual server is currently available to accept connections. If you omit this parameter, the 3-DNS uses the prober <ip_addr> parameter defined in the globals statement. This applies to hosts only.
port	Specifies the port used to probe this host if the probe_protocol parameter is set to TCP. This applies to hosts only.

Table A.18 Address information sub-statements

Limit settings

Using the **limit** sub-statement, you can manage the physical and throughput resources of your BIG-IP systems, EDGE-FX Caches, hosts, and their respective virtual servers. If you omit this sub-statement, the 3-DNS does not use resource thresholds to monitor the availability of the BIG-IP systems, EDGE-FX Caches, or hosts, and their respective virtual servers.

Parameter	Description
limits	Indicates the start of the limits definition. Applies to BIG-IP systems and their virtual servers, EDGE-FX Caches and their virtual servers, and hosts and their virtual servers.
cpu_avail	Specifies, in percentage, how much CPU processing must remain available on the server or virtual server. The cpu_avail parameter applies to hosts and EDGE-FX Caches only.
mem_avail	Specifies, in kilobytes, how much memory must remain available on the server or virtual server. The mem_avail parameter applies to hosts and EDGE-FX Caches only.
disk_avail	Specifies, in kilobytes, how much disk space must remain available on the server or virtual server. The disk_avail parameter applies to hosts and EDGE-FX Caches only.
kbytes_per_sec	Specifies, in kilobytes per second, the maximum allowable throughput rate for the server or virtual server.
pkts_per_sec	Specifies, in packets per second, the maximum allowable data transfer rate for the server or virtual server
current_conn	Specifies the maximum number of current connections for the server or virtual server.

Table A.19 Limit sub-statement

Remote connections

You use the **remote** sub-statement only if you want to specify a different login name, or specifically use SSH or RSH, on 3-DNS systems, BIG-IP systems, EDGE-FX Caches, or GLOBAL-SITE systems.

Parameter	Description
remote	Indicates the start of a remote sub-statement. Applies to any of the following server types: 3-DNS, BIG-IP, GLOBAL-SITE, or EDGE-FX Cache.
secure	Specifies whether to use SSH (secure shell) or RSH (remote shell) for remote connections. The default for crypto systems is yes , which specifies that SSH is used. Non-crypto versions must use RSH instead. Applies to 3-DNS systems, BIG-IP systems, and EDGE-FX Caches. (Note that GLOBAL-SITE systems are only available as crypto systems.)
user	Specifies the "superuser" name that is used to allow a remote user to log on to the system. Enclose this name in quotation marks. If you omit this parameter, the default, " root ", is used. Applies to any of the following server types: 3-DNS, BIG-IP, GLOBAL-SITE, or EDGE-FX Cache.

Table A.20 Remote connections sub-statements

Hardware redundancy

If you have hardware-redundant 3-DNS or BIG-IP systems, you must configure the **interface** sub-statement so that the 3-DNS works properly with BIG-IP redundant systems running in Active-Active mode. This sub-statement is also required in using the standby BIG-IP or 3-DNS for probing.

Parameter	Description
interface	Indicates the start of the interface sub-statement.
address	Specifies the IP address of both network interface cards, on separate lines. Applies to 3-DNS systems and BIG-IP systems.

Table A.21 Hardware redundancy sub-statements

Factories

For any 3-DNS, BIG-IP, GLOBAL-SITE, and EDGE-FX Cache system, you can change the number and types of probing factories by using the **factories** sub-statement. If you omit this sub-statement, the 3-DNS uses the defaults settings specified in the **globals** statement. For more information on factories and probing, see Chapter 4, *The big3d Agent*.

Parameter	Description
factories	Indicates the start of the factories definition. Applies to 3-DNS, BIG-IP, EDGE-FX Cache, and GLOBAL-SITE systems.
prober	Specifies the number of prober factories to use. The default setting is 5.
snmp	Specifies the number of SNMP factories to use. Note that you must use an SNMP factory to collect metrics from an EDGE-FX Cache. The default setting is 1.
hops	Specifies the number of hops factories to use. The default setting is 0.
ecv	Specifies the number of ECV factories to use. The default setting is 5.

Table A.22 *Factories sub-statements*

SNMP settings

The **snmp** sub-statement is valid for hosts and EDGE-FX Caches only. This sub-statement instructs the **big3d** agent to use an SNMP agent on the host or the cache to collect additional metrics information.

If you need help configuring the SNMP agent on the EDGE-FX Cache, refer to the *EDGE-FX Administrator Guide*. If you need help configuring the SNMP agent on the host, refer to the documentation provided with the host.

Parameter	Description
snmp	Specifies the start of an SNMP definition. Applies to hosts and EDGE-FX Caches only.
agent	Specifies the SNMP agent type. If you omit this parameter, the big3d agent uses the generic SNMP agent. Applies to hosts and EDGE-FX Caches only.
port	Specifies the port the SNMP agent runs on. Applies to hosts and EDGE-FX Caches only.
community	Specifies the password for basic SNMP security and for grouping SNMP hosts. Enclose this string in quotation marks. Applies to hosts and EDGE-FX Caches only.
timeout	Specifies the amount of time (in seconds) for the timeout. The default is appropriate in most cases. If you are contacting a host through a very slow network, you can try increasing the timeout and retries values to improve performance. However, the problem with increasing these values is that a host that is down may hold up other SNMP responses for an excessive amount of time. Applies to hosts only.

Table A.23 *SNMP sub-statements*

Parameter	Description
retries	Specifies the number of times requests should be retried. The default is appropriate in most cases. If you are contacting a host through a very slow network, you can try increasing the timeout and retries values to improve performance. However, the problem with increasing these values is that a host that is down may hold up other SNMP responses for an excessive amount of time. Applies to hosts only.
version	Specifies the SNMP agent version number. Applies to hosts only.

Table A.23 *SNMP sub-statements*

Virtual server definitions

Part of defining a BIG-IP, EDGE-FX Cache, or host is defining the virtual servers that the server manages. You can then use the virtual servers that you define as part of the **server** statement in a **wideip** definition for load balancing.

Parameter	Description
vs	Indicates the start of a virtual server definition.
address	Specifies the IP address of the virtual server. Note that the virtual server's address must be listed first, before port or service values.
port or service	Specifies the virtual server's port number or service name. You can add the port number, preceded by a colon, on the same line as the virtual server's address, or you can enter it on the next line. You can use the service name if it is a WKS (well known service) and you enclose it in quotation marks.
limit	Specifies resource thresholds for the virtual server. Note that if a virtual server reaches a limit, the virtual server is marked as unavailable for load balancing.
depends_on	Specifies the IP address and port of other virtual servers that must also be available for load balancing (up status) before the 3-DNS uses this virtual server for load balancing.
probe_protocol	Specifies the protocol to use for probing this virtual server: ICMP or TCP.
translate	Specifies that iQuery packets sent to the BIG-IP include translated IP addresses (required if the packets must pass through a firewall). When you use this keyword, you must then include address and port/service information for the translated IP addresses. Applies to BIG-IP virtual servers only.

Table A.24 *Virtual server definitions*

The datacenter statement

A **datacenter** statement defines the group of 3-DNS systems, BIG-IP systems, EDGE-FX Caches, GLOBAL-SITE systems, and hosts that reside in a single physical location.

Syntax for the datacenter statement

The **datacenter** statement uses the following syntax.

```
datacenter {
  name <"data center name">
  [ location <"location info"> ]
  [ contact <"contact info"> ]
  [ 3dns <IP address | name> ]
  [ bigip <IP address | name> ]
  [ edgefx <IP address | name> ]
  [ gsite <IP address | name> ]
  [ host <IP address | name> ]
}
```

Figure A.14 Syntax for the *datacenter* statement

Figure A.15 shows an example of a valid **datacenter** statement.

```
datacenter {
  name "New York"
  location "NYC"
  contact "3DNS_Admin"
  3dns 192.168.101.2
  bigip 192.168.101.40
  edgefx 192.168.101.50
  gsite 192.168.101.70
  host 192.168.105.40
}
```

Figure A.15 Example syntax for the *datacenter* statement

Definition of datacenter sub-statements

The **datacenter** sub-statements specify a name for the data center and the machines it contains.

Parameter	Description
name	Specifies the name of this data center. The name must be enclosed in quotation marks.
location	Specifies the location of the data center. This name must be enclosed in quotation marks. This sub-statement is not required, but this information can be useful if problems later arise or changes are required.
contact	Identifies the administrator of the data center. This name must be enclosed in quotation marks. This sub-statement is not required, but this information can be useful if problems later arise or changes are required.
3dns	Specifies the IP address of a 3-DNS in this data center.
bigip	Specifies the IP address of a BIG-IP in this data center.
edgefx	Specifies the IP address of an EDGE-FX Cache in this data center.

Table A.25 Data center sub-statements

Parameter	Description
gsite	Specifies the IP address of a GLOBAL-SITE in this data center.
host	Specifies the IP address of a host in this data center.

Table A.25 Data center sub-statements

The sync_group statement

The **sync_group** statement defines the group of 3-DNS systems that synchronize their configuration settings and metrics data. You configure this statement in the **wideip.conf** file of the principal 3-DNS.

Syntax for the sync_group statement

The **sync_group** statement uses the following syntax.

```
sync_group {
    name <"name">
    3dns <ip_address | "name">
    [ 3dns <ip_address | "name"> ]
}
```

Figure A.16 Syntax for the sync_group statement

Note that the **sync_group** statement does not support location or contact sub-statements.

Figure A.17 shows an example of a valid **sync_group** statement.

```
sync_group {
    name "sync"
    3dns 192.168.101.2 // New York - this is the principal system
    3dns 192.168.102.2 // Los Angeles - this is a receiver system
    3dns 192.168.103.2 // Madrid - this is also a receiver system
}
```

Figure A.17 Example syntax for the sync_group statement

Definition of sync_group sub-statements

The **sync_group** sub-statements define the members of the sync group.

Parameter	Description
name	Specifies the name of this sync group.
3dns	Specifies the IP address or domain name (enclosed in quotation marks) of a 3-DNS in the group. First list the IP address of the principal system. Then list all other 3-DNS systems, in the order that they should become a principal system, if the previously listed principal 3-DNS fails. Note that there can only be one principal system in a sync group at any time.

Table A.26 Sync_group sub-statements

The wide IP statement

The **wideip** statement defines a wide IP. A **wide IP** maps a domain name to a load balancing mode and a set of virtual servers.

Syntax for the wideip statement

The **wideip** statement uses the following syntax.

```
wideip {
  address <ip_address>
  port <port_number> | <"service name">
  [ ttl <number> ]
  [ persist < yes | no > ]
  [ persist_ttl <number> ]
  name <"domain_name">
  [ alias <"alias_name"> ... ]
  [ port_list <port_number> <port_number> ... ]
  [ qos_coeff {
    rtt <number>
    hops <number>
    completion_rate <number>
    packet_rate <number>
    vs_capacity <number>
    topology <number>
    kbps <number>
  } ]
  [ pool_lbmode <rr | ratio | ga | random | topology> ]
  [ ecv {
    protocol <none | ftp | http | https>
    file_name <string>
    user <string>
    password <string>
    hashed_password <string>
    scan_level <none | all | first>
    transfer_amount <number>
    connection_timeout <number>
    transfer_timeout <number>
    search_string <"string">
  } ]
  pool {
    name <"pool_name">
    [ ttl <number> ]
    [ ratio <number> ]
    [ last_resort <yes | no> ]
    [ check_static_depends < yes | no > ]
    [ check_dynamic_depends < yes | no > ]
    [ limit {
      [ kbytes_per_sec <number> ]
      [ pkts_per_sec <number> ]
      [ current_conns <number> ]
      [ cpu_avail <number> ]
      [ disk_avail <number> ]
      [ mem_avail <number> ]
    } ]
  }
  [ type < A | CNAME >
    [ cname <canonical name> ]
```

Figure A.18 Syntax for the **wideip** statement

```

[ dynamic_ratio < yes | no > ]
[ rr_ldns < yes | no > ]
[ rr_ldns_limit <number> ]
[ preferred < completion_rate | ga | hops | leastconn | packet_rate | qos | random |
ratio | return_to_dns | rr | rtt | topology | vs_capacity | null | static_persist |
kpbs>]
[ alternate < ga | null | random | ratio | return_to_dns | rr | topology | packet_rate
| leastconn | vs_capacity | static_persist > ]
[ fallback < completion_rate | ga | hops | leastconn | packet_rate | qos | random |
ratio | return_to_dns | rr | rtt | topology | vs_capacity | null | static_persist |
kpbs > ]
    address <vs IP address:[port]> [ratio <weight>
    ...]
    }
}

```

Figure A.18 Syntax for the *wideip* statement

Figure A.19 shows an example of a valid **wideip** statement.

```

wideip {
    address          192.168.102.50
    service          "http"
    name             "http.wip.domain.com"
    alias            "store.wip.domain.com"
    alias            "*.wip.domain.com"
    alias            "http.wip.domain.???"
    pool_lbmode      ratio
    pool {
        name         "pool_1"
        ratio        3
        limit {
            kbytes_per_second 10000
        }
        preferred    rtt
        alternate    random
        address      192.168.101.50
        address      192.168.102.50
        address      192.168.103.50
    }
    pool {
        name         "pool_2"
        ratio        1
        limit {
            kbytes_per_second 10000
        }
        preferred    ratio
        address      192.168.104.50    ratio 2
        address      192.168.105.50    ratio 1
    }
}

```

Figure A.19 Example syntax for the *wideip* statement

Definition of wideip sub-statements

The **wideip** sub-statements define groups of virtual servers to be load balanced, and they assign load balancing characteristics, such as the load balancing mode, to each group.

Address information

The address information sub-statements specify the IP address, name, and alias of the wide IP. They also specify the pool of virtual servers that the wide IP load balances.

Parameter	Description
address	Specifies the IP address registered with InterNIC that corresponds to the wide IP name. This IP address is also listed as the A record in the zone file for the domain.
port or service	Specifies the default port number or service name for the wide IP. You can use the service name if it is a well known service (WKS) and you enclose it in quotation marks.
name	Specifies the fully qualified domain name for the wide IP (for example, " www.wip.domain.com "). You must enclose all names in quotation marks. Note that you can use two wildcard characters, the asterisk (*) and the question mark (?), in wide IP names. The asterisk (*) can represent multiple characters, and the question mark (?) can represent a single character. Any of the following examples are valid for the name or alias parameter in a wideip statement: " www.*.com ", " *.domain.com ", " *.domain.??? ", and so on.
alias	Specifies an alternate name for the wide IP. The conventions for name also apply to alias . You can specify an unlimited number of alias names for each wide IP.

Table A.27 Address information sub-statements

Load balancing sub-statements

The load balancing sub-statements denote the general load balancing attributes for all pools in the **wideip.conf** file.

Parameter	Description
ttl	Specifies the amount of time (in seconds) that the A record is used by the LDNS after resolving the wide IP. This is the TTL associated with the A record as specified by RFC 1035.
persist	Specifies whether to maintain a persistent connection between an LDNS and a particular virtual server in the wide IP (rather than load-balancing the connection to any available virtual server). Note that the variables drain_requests and default_persist_ttl , in the globals statement, affect this setting. See page A-10 for more information.
persist_ttl	Specifies the number of seconds to maintain a persistent connection between an LDNS and a particular virtual server in this wide IP; this setting is valid only if you have configured the persist parameter.
port_list	Specifies a list of ports that must be available before the 3-DNS can send connections to the specified address.
qos_coef	Specifies the relative weighting for each load balancing method in calculating the Quality of Service mode. Before you adjust any QOS coefficients, you may want to review Chapter 9, <i>Working with Quality of Service</i> , in the 3-DNS Administrator Guide .
pool_lbmode	Specifies the load balancing mode to use to balance requests over all pools.

Table A.28 Load balancing sub-statements

ECV sub-statements

The ECV sub-statements define the components of an extended content verification (ECV) monitor. Use the ECV sub-statement if you want the 3-DNS to verify the presence of a file, or certain content, on the servers or virtual servers that host the content mapped to the wide IP, before the wide IP is considered **up** for load balancing.

Parameter	Description
ecv	Specifies an extended content verification (ECV) monitor for a virtual server in a pool.
protocol	Specifies the protocol to use for the ECV. You can use only http , https , or ftp . Use only with the ecv sub-statement.
file_name	Specifies the name of the object to retrieve. Use only with the ecv sub-statement.
user	Specifies the user name that you use to log in to the service. Use only with the ecv sub-statement.
password	Specifies the password that corresponds to the user account. Use only with the ecv sub-statement.
hashed_password	Specifies the password in encrypted characters. Use only with the ecv sub-statement.
scan_level	Specifies whether you want to scan just through the configured wide IP names, or through the wide IP names and aliases. Use only with the ecv sub-statement. Note that if you use wildcard characters in the wide IP name or alias parameters, those names and aliases are ignored by the ECV scans.
transfer_amount	Specifies the number of bytes to transfer. Use only with the ecv sub-statement.
transfer_timeout	Specifies the maximum amount of time the file information transfer should take. Use only with the ecv sub-statement.
connection_timeout	Specifies the maximum amount of time to connect to a service. Use only with the ecv sub-statement.
search_string	Specifies a regular expression that you want the ECV monitor to locate within the scanned file.

Table A.29 ECV sub-statements

Pool sub-statements

The **pool** sub-statements define the virtual servers, and the load balancing modes within the pool, that the 3-DNS uses to respond to DNS requests. Note that you can have one or more pools in a wide IP definition.

Parameter	Description
pool	Indicates the start of the pool definition for this wide IP. A pool is a set of virtual servers defined and owned by a BIG-IP, EDGE-FX Cache, or host machine.
name	As part of a pool definition, defines the name of the pool. All names must be enclosed in quotation marks.
tll	Specifies the amount of time (in seconds) that the A record is used by the LDNS after resolving the wide IP. This is the TTL associated with the A record as specified by RFC 1035.
ratio	As part of a pool definition, ratio specifies the default weighting to use, with respect to other pool types, when the pool_lbmode is ratio .
last_resort	Specifies whether the 3-DNS directs LDNS requests to this pool when no other pools in the wide IP successfully respond to the request. The default setting is no .
check_static_depends	Specifies whether the 3-DNS checks availability before returning a virtual server in the pool. (Note that this parameter does not affect the status of the virtual server on the Virtual Server Statistics screen, in the Configuration utility, while the global variable of the same name does affect the status.)
check_dynamic_depends	Specifies whether the 3-DNS checks paths before returning a virtual server in the pool.
type	Specifies the type of pool. The default is A . You can also use CNAME to redirect LDNS requests to a CDN provider in the cdn.inc file.
cname	Specifies the canonical name (cname) for the pool. Use this attribute with the pool type CNAME to redirect LDNS requests to a name server in another network, or to a CDN provider. Enclose the cname in quotation marks.
dynamic_ratio	Specifies whether the 3-DNS treats QOS scores as ratios, and uses each server in proportion to the ratio determined by the QOS calculation. The default is no .
rr_ldns	Specifies whether the 3-DNS returns a list of available virtual servers available for load balancing to a client and stores the list in the browser cache. The default is no , which specifies that the 3-DNS returns only one A record per query.
rr_ldns_limit	The maximum number of A records to return when rr_ldns is set to yes . You can enter a value between 0 and 16 . The default is 0 , which specifies that the 3-DNS returns the IP addresses of all (up to 16) available virtual servers.
preferred	Specifies the load balancing mode to use for the specified pool. Each acceptable value is described in the next table. The default is rr (Round Robin).
alternate	Specifies the load balancing mode to use for the specified pool if the preferred mode fails. The default is rr (Round Robin). Also see the description of default_alternate in Table A.12, on page A-15.

Table A.30 Pool sub-statements

Parameter	Description
fallback	Specifies the load balancing mode to use for the specified pool if the alternate mode fails. If the fallback mode fails, the 3-DNS returns the request to DNS. The default is return_to_dns . Also see the description of default_fallback in Table A.12, on page A-15
address	As part of a pool definition, address specifies the IP address of each virtual server in the pool. You can use the same virtual server in multiple pools, but not within the same pool.
port	Specifies a specific port to use for the specified virtual server. This sub-statement is optional. A port specified here overrides the wide IP's port setting. If a port is not specified here, the wide IP's port value is assumed.
ratio	As part of a virtual server's address specification, ratio defines the default weighting to use with respect to all virtual servers in this pool when the Ratio load balancing mode is employed. The default is 1.

Table A.30 Pool sub-statements

Load balancing modes

The load balancing sub-statements specify the load balancing modes to use for the wide IP in this order:

- The 3-DNS attempts to load balance requests using the **preferred** mode.
- If the **preferred** mode fails, the 3-DNS tries the **alternate** mode.
- If the **alternate** mode fails, the 3-DNS tries the **fallback** mode.
- If the **fallback** mode fails, the request is returned to DNS.
DNS attempts to resolve the request based on the contents of the zone files.

As noted in Table A.31, not all modes are valid for the **alternate** sub-statement. Also note that the **alternate** and **fallback** sub-statements accept two additional values, **return_to_dns** and **null**.

If you do not specify a load balancing mode within a pool, the wide IP uses the default load balancing mode defined in the **globals** statement (see page A-5).

Parameter	Description
completion_rate	Sends each new connection to the server that has the fewest number of dropped packets. Valid in a preferred or fallback sub-statement.
global_availability (ga)	Distributes connections to a list of servers, always sending a connection to the first available server in the list.
hops	Sends each new connection to the server that has the fewest number of network hops between the server and the client LDNS. Valid in a preferred or fallback sub-statement.
leastconn	Sends each new connection to the server that currently hosts the fewest current connections.

Table A.31 Load balancing mode sub-statements

Parameter	Description
null	Bypasses the current load balancing method and forces the 3-DNS to use the next load balancing method or, if it has cycled through all load balancing sub-statements for the pool, to the next pool. Valid in an alternate or fallback sub-statement.
packet_rate	Sends each new connection to the server that is managed by a BIG-IP currently handling the least amount of network traffic (determined by the fewest number of packets currently processed by the system).
qos	Takes these performance factors into account when determining how to distribute connections: hops, packet rate, completion rate, round trip time, kbps, virtual server capacity, and topology. You can configure how much emphasis to place on each performance factor, or you can configure the Quality of Service mode to treat all factors as being equally important. Valid in a preferred or fallback sub-statement.
random	Distributes each new connection to a server chosen at random from the wide IP set of virtual servers.
ratio	Distributes new connections across servers in proportion to a user-defined ratio.
return_to_dns	Returns the resolution request to DNS, preventing the 3-DNS from using the next load balancing method or using the next available pool.
rr	Distributes connections evenly across all servers, passing each new connection to the next virtual server in line.
rtt	Sends each new connection to the server that demonstrates the fastest round trip time between the server and the client LDNS. Valid in a preferred or fallback sub-statement.
topology	Distributes connections based on the proximity of an LDNS to a particular data center. You must also configure a topology statement before this load balancing mode works.
static_persist	Distributes connections to a virtual server based on IP address only. The 3-DNS always returns the same virtual server to the same client, if the virtual server is available.
vs_capacity	Distributes connections based on the overall available capacity of the virtual server. Over time all virtual servers in the pool receive connections, but the virtual server with the most capacity receives the highest percentage of connections.
kbps	Distributes connections to the virtual server with the lowest kilobytes per second throughput rate.

Table A.31 Load balancing mode sub-statements

Use the following equation to configure the Quality of Service load balancing mode:

$$A (1/\text{packet rate}) + B (1/\text{rtt}) + C (\text{completion rate}) + D (\text{topology}) + E (1/\text{hops}) + F (1/\text{kbps}) + G (\text{vs_capacity})$$

◆ **Note**

For more information about load balancing modes, see Chapter 8, **Load Balancing**.

The topology statement

The **topology** statement implements a form of wide-area IP filtering, based on the geographic attributes of the DNS message. For example, you can specify that requesting LDNS clients in North America are allowed access to data centers in North America, but not allowed access to data centers in South America.

By including a **topology** statement in your **wideip.conf** file, you can use the topology load balancing mode, both on its own and as part of the Quality of Service mode.

For more information on using the Topology load balancing mode, see Chapter 7, *Configuring a Globally Distributed Network*, and Chapter 8, *Configuring a Content Delivery Network*, in the **3-DNS Administrator Guide**. For more information on topology in general, see Chapter 13, *Topology*, in this guide.

Syntax for the topology statement

Figure A.20 contains examples of the syntax used in the **topology** statement. Note that the object names are in quotation marks.

```

topology {
    longest_match <yes | no>

    // server          ldns          score
    pool.<"pool_name">  cont.<"continent_name">  <number>
    datacenter.<"dc_name"> !country.<"2-letter_code"> <number>
    pool.<"pool_name">  user.<"region_name">  <number>
    pool.<"pool_name">  isp."AOL"  <number>
}

```

Figure A.20 Syntax for the **topology** statement

◆ Note

In a topology statement, use the **not** operator (!) to negate the meaning of an element, as shown in the example in Figure A.20.

Definition of topology sub-statements

The topology sub-statements define the topology records that the 3-DNS uses for Topology load balancing.

Parameter	Description
longest_match	In cases where there are topology records that match a particular IP address, longest_match specifies whether the 3-DNS selects the record that is most specific, and thus has the longest match. When longest_match is set to yes , the topology records are sorted according to the longest match criteria.
server	Specifies the location of the virtual servers.

Table A.32 Topology sub-statements

Parameter	Description
ldns	Specifies the location of the LDNS making the name resolution request.
pool.<"pool_name">	Specifies a wide-IP pool for load balancing. Note that pool names can be duplicated across wide IPs. The name must be in quotation marks. Use this for server in a topology record.
datacenter.<"datacenter_name">	Specifies a data center for load balancing. The name must be in quotation marks. Use this for server in a topology record.
continent.<"continent_name">	Specifies one of the continents for load balancing: " North America ", " South America ", " Europe ", " Asia ", " Australia ", " Africa ", or " Antarctica ". The name must be in quotation marks. Use this for ldns in a topology record.
country.<"2-letter_code">	Specifies a country for load balancing using one of the two-letter country codes found in the file <code>/var/3dns/include/net.ccdb</code> . The name must be in quotation marks. Use this for ldns in a topology record.
isp."AOL"	For local DNS servers only, specifies the Internet service provider, America Online (AOL). The name must be in quotation marks.
user.<"region_name">	Specifies a user-defined region. The name must be in quotation marks.
!	The not (!) operator negates the meaning of an element in a topology record.
score	Specifies the relative weight, or score, for the topology record, which allows the 3-DNS to evaluate the best resolution option for a DNS request.

Table A.32 Topology sub-statements

Access control lists

You can now create access control lists (ACLs) that contain a group of LDNS IP addresses whose paths the 3-DNS will not probe. The two types of ACLs are:

- Prober
- Hops

Syntax for the access control lists

The access control lists use the following syntax.

```
actions {
  NO_RELAY
  delete rdb ACL region "probe_acl"
  delete rdb ACL region "hops_acl"
}
region_db ACL {
  region {
    name "probe_acl"
    <ldns cidr>
    <ldns cidr>
  }
  region {
    name "hops_acl"
    region "probe_acl"
    <ldns cidr>
    <ldns cidr>
  }
}
```

Figure A.21 Syntax for the access control lists

Definition of the access control list sub-statements

The access control list sub-statements define local DNS servers that should not be probed.

Parameter	Description
actions	Include, but do not modify this sub-statement.
region_db ACL	Specifies that ACLs are being created.
region	Specifies groups of CIDRs by probe type.
name	Specifies the name of the ACL.
probe_acl	The 3-DNS restricts any big3d agent from probing the defined group of local DNS servers.
hops_acl	The 3-DNS restricts any big3d agent from tracerouting the defined group of local DNS servers

Table A.33 Access control list sub-statements

◆ Note

For more information on ACLs, refer to Chapter 3, **Access Control Lists**.

Working with comments

You can insert comments anywhere you would otherwise see white space in the 3-DNS configuration file.

Syntax

Note that the comment syntax depends on the environment in which you use the configuration file.

```
/* This is a 3-DNS comment as in C */  
// This is a 3-DNS comment as in C++  
# This is a 3-DNS comment as in common UNIX shells and Perl
```

Figure A.22 Syntax for comments

Definition and usage

The format for comments varies by programming language; each format is described below. To avoid comment nesting problems, we recommend that you use only one comment style in your **wideip.conf** file. However, all styles may be used in a single **wideip.conf** file.

C style comments

C style comments start with the slash character, followed by the asterisk character (`/*`), and end with the asterisk character, followed with the slash character (`*/`). Because the comment is completely delimited with these characters, a comment can span multiple lines.

Note that C style comments cannot be nested. For example, the following syntax is not valid because the entire comment ends with the first `*/`.

```
/* This is the start of a comment.  
   This is still part of the comment.  
/* This is an incorrect attempt to nest a comment. */  
   This is no longer in any comment. */
```

Figure A.23 Syntax for C style comments

C++ style comments

C++ style comments start with two slash characters (`//`) and are no longer than one line in length. To have one logical comment span multiple lines, each line must start with the `//` pair.

```
// This is the start of a comment. The next line  
// is a new comment line, even though it is  
// logically part of the previous comment.
```

Figure A.24 Syntax for C++ style comments

Shell style comments


Shell style (also known as Perl style) comments start with the number character (#) and are no longer than one line in length.

```
# This is the start of a comment. The next line
# is a new comment line, even though it is logically
# part of the previous comment.
```

Figure A.25 Syntax for shell style comments

Understanding current values

You may notice several current values in the **wideip.conf** file. Current values are preceded by the **cur_** prefix in the **wideip.conf** file. The purpose of current values is to pre-load the database with previously collected statistics and metrics. The collected statistics and metrics are useful if you want to quickly restart a 3-DNS without a temporary loss of intelligence.

You may notice current values associated with **server**, **vs**, **path**, or **wideip** definitions. (You can also view current values by clicking the Configuration View button  in the Configuration utility.) The current values parameters show the real-time status of the servers, virtual servers, local DNS server paths, and wide IPs that make up your configuration. Examples of current values for each type of definition follow.

◆ WARNING

Do not edit the current values statements unless you are a very experienced 3-DNS user, or you are instructed to do so by your vendor.

Server definition current values

Server definitions may contain several current values, as shown in Figure A.26.

```
// New York BIG-IP
server {
  type bigip
  address 192.168.101.40
  cur_ok 1 //Up
  cur_packet_rate 6
  cur_packet_in 1872
  cur_packet_out 1812
  cur_uptime 3615 //60 mins 15 Secs
  [virtual server definitions]
}
```

Figure A.26 Example of current values in a server definition

The current values parameters that are shown in Figure A.26 are defined in Table A.34. Note that you may see more current values than those listed here.

Parameter	Description
cur_ok	Indicates the state of the specified server. The options are: 1 (up), 2 (down), 3 (waiting), 4 (alert), and 5 (panic).
cur_packet_rate	Indicates the number of packets per second sent during the last sample period.
cur_packet_in	Indicates the number of packets that the server has received.
cur_packet_out	Indicates the number of packets the server has sent.
cur_uptime	Indicates the length of time that the server has been running since the last reboot.

Table A.34 Description of current values in a server definition

Virtual server definition current values

Virtual server definitions may contain several current values, as shown in Figure A.27.

```
vs {
  address 192.168.102.50:80 //http
  [ depends_on {
    address 109.168.102.50:20 //ftp-data
    address 192.168.102.50:443 //https
  } ]
  limit { /* none */ }
  probe_protocol tcp
  cur_state 1 // green
  cur_nodes_up 3
  cur_connections 0
  ...
  cur_picks 0
  cur_refreshes 41
}
```

Figure A.27 Example of current values in a virtual server definition

The current values parameters that are shown in Figure A.27 are defined in Table A.35. Note that you may see more current values than those listed here.

Parameter	Description
cur_state	Indicates the availability of the virtual server to receive connection requests. The options are: 1 (green - available), 2 (red - down), 3 (blue - unknown), 4 (yellow - unavailable)
cur_nodes_up	Indicates the number of active servers serving the specified virtual server.
cur_connections	Indicates the number of connections to the specified virtual server.
cur_picks	Indicates the number of times the specified virtual server was returned by the 3-DNS.
cur_refreshes	Indicates the number of times the server and connection counts were refreshed with new data.

Table A.35 Description of current values in a virtual server definition

Local DNS server paths current values

Path definitions for local DNS servers may contain several current values, as shown in Figure A.28.

```
path {
  address 10.25.50.100 // LDNS
  cur_rtt 102382
  cur_completion_rate 10000
  cur_picks 239
  cur_accesses 302
}
```

Figure A.28 Example of current values in a path definition

The current values parameters that are shown in Figure A.28 are defined in Table A.36. Note that you may see more current values than those listed here.

Parameter	Description
cur_rtt	Indicates the round trip time (RTT), which is a calculation of the time (in microseconds) that the specified machine takes to respond to a probe issued by the 3-DNS.
cur_completion_rate	Indicates the percentage of completed packets versus lost packets, using this equation: [1 - (packets received / sent)] X 10000.

Table A.36 Description of current values in a path definition

Parameter	Description
cur_picks	Indicates the number of times this path's data resulted in the virtual server being chosen for a connection. This only applies if a wide IP is doing dynamic load balancing (using path data).
cur_accesses	Indicates the number of times this path was considered when performing dynamic load balancing.

Table A.36 Description of current values in a path definition

Wide IP definition current values

Wide IP definitions may contain several current values, as shown in Figure A.29.

```
wideip {  
  address 192.168.102.70  
  name "www.domain.com"  
  port 80  
  cur_preferred 143982  
  cur_alternate 108090  
  cur_fallback 130094  
  cur_returned_to_dns 23872  
  [pool definitions]  
}
```

Figure A.29 Example of current values in a wide IP definition

The current values parameters that are shown in Figure A.29 are defined in Table A.37. Note that you may see more current values than those listed here.

Parameter	Description
cur_preferred	Indicates the number of times the specified wide IP was resolved by the preferred load balancing mode.
cur_alternate	Indicates the number of times the specified wide IP was resolved by the alternate load balancing mode.
cur_fallback	Indicates the number of times the specified wide IP was resolved by the fallback load balancing mode.
cur_returned_to_dns	Indicates the number of times the specified wide IP did not find a suitable virtual server to return using the preferred , alternate , or fallback load balancing modes. In this situation, the 3-DNS returns the wide IP key (fallback address) as specified in the zone file.

Table A.37 Description of current values in a wide IP definition

◆ **Tip**

*To find out how many times the 3-DNS has received resolution requests for a wide IP, add the values for **cur_preferred**, **cur_alternate**, and **cur_fallback**.*



B

3dpipe Command Reference

3dpipe commands

The **3dpipe** utility is a command line utility that you can use to view summary information, and to enable and disable several objects in the 3-DNS configuration. This chapter lists the various **3dpipe** commands and their syntax requirements. Table B.1 outlines the conventions used in the command line syntax.

Item in text	Description
\	Continue to the next line without typing a line break.
< >	You enter text for the enclosed item. For example, if the command has <wideip_name> , type the name of the wide IP.
" "	Names that have spaces in them must be enclosed in quotation marks.
	Separates alternate options for a command.
[]	Syntax inside the brackets is optional.
...	Indicates that you can type a series of items.

Table B.1 *Command line conventions*

Table B.2 provides a concise listing of the individual **3dpipe** commands, along with the page reference where you can find a detailed description of the command syntax and usage.

Command	Description	Page
datacenter (or dc)	Displays all data centers, and data center status.	B-3
-help (or -h)	Displays online help for 3dpipe command syntax.	B-4
<server type>	Displays all servers of the specified type, the status for servers of the specified type, and all virtual servers for servers of the specified type	B-5
stats	Displays summary statistics for the 3-DNS.	B-6
syncgroup (or sg)	Displays the sync group servers and the sync group status.	B-7
-version (or -v)	Displays the version number for the portal.	B-8

Table B.2 *Summary of the 3dpipe commands*

Command	Description	Page
virtual (or vs)	Displays virtual server status.	B-9
wideip (or wip)	Displays all wide IPs, wide IP pools, wide IP status, and wide IP pool status	B-10

*Table B.2 Summary of the **3dpipe** commands*

datacenter (or dc)

```
3dpipe datacenter show all
3dpipe dc <datacenter_name> status
3dpipe dc <datacenter_name> disable [<time in seconds> [<by_whom>]]
3dpipe dc <datacenter_name> enable
```

With the **datacenter** command, you can perform the following tasks:

- View all the configured data centers
- View whether a specific data center is enabled or disabled (the status)
- Disable a specific data center for a certain period of time
- Enable a data center that is disabled

-help (or -h)

```
3dpipe [-h | -help]
```

The **-help** or **-h** flag displays the **3dpipe** command syntax or usage text for all available commands.

◆ **Note**

*To display detailed online help for the **3dpipe** command, type: **man 3dpipe**.*

<server type>

```
3dpipe <3dns | bigip | edgefx | edge_fx | globalsite | gsite | host> show all
3dpipe <3dns | bigip | edgefx | edge_fx | globalsite | gsite | host> <ip_address> status
3dpipe <3dns | bigip | edgefx | edge_fx | globalsite | gsite | host> <ip_address>
    [disable <time in seconds> [<by_whom>]]
3dpipe <3dns | bigip | edgefx | edge_fx | globalsite | gsite | host> <ip_address> enable
3dpipe <bigip | edgefx | edge_fx | host> <ip_address> virtual show all
```

With the **<server type>** command, you can perform the following tasks:

- View all the servers of the specified type
- View whether a specific server is enabled or disabled (the status)
- Disable a specific server for a certain period of time
- Enable a server that is disabled
- View the virtual servers for a specific server

stats

```
3dpipe stats summary
```

With the **stats** command, you can view a summary of the real-time statistics for the 3-DNS.

syncgroup (or sg)

```
3dpipe syncgroup [<syncgroup_name>] show servers
3dpipe sg [<syncgroup_name>] status
3dpipe sg [<syncgroup_name>] disable [<time in seconds> [<by_whom>]]
3dpipe sg [<syncgroup_name>] enable
```

With the **syncgroup** command, you can perform the following tasks:

- View all the members in a sync group
- View whether a specific sync group is enabled or disabled (the status)
- Disable a specific sync group for a certain period of time
- Enable a sync group that is disabled

-version (or -v)

```
3dpipe <-version | -v>
```

The **version** command displays the current version of the iControl portal that is used by the **3dpipe** utility.

virtual (or vs)

```
3dpipe virtual <ipaddress>:<port> status
3dpipe vs <ipaddress>:<port> disable [<time in seconds> [<by_whom>]]
3dpipe vs <ipaddress>:<port> enable
```

With the **virtual** command, you can perform the following tasks:

- View whether a specific virtual server is enabled or disabled (the status)
- Disable a specific virtual server for a certain period of time
- Enable a virtual server that is disabled

wideip (or wip)

```
3dpipe wideip show all
3dpipe wip <wideip_name> status
3dpipe wip <wideip_name> disable [<time in seconds> [<by_whom>]]
3dpipe wip <wideip_name> enable
3dpipe wip <wideip_name> pool show all
3dpipe wip <wideip_name> pool <pool_name> status
3dpipe wip <wideip_name> pool <pool_name> disable [<time in seconds> [<by_whom>]]
3dpipe wip <wideip_name> pool <pool_name> enable
3dpipe wideip <wide_IP_name> pool <pool_name> virtual show all
3dpipe wideip <wide_IP_name> dc <datacenter_name> disable [<time in seconds>
    [<by_whom>]]
3dpipe wideip <wide_IP_name> dc <datacenter_name> enable
3dpipe wideip <wide_IP_name> dc <datacenter_name> status
```

With the **wideip** command, you can perform the following tasks:

- View all wide IPs
- View whether a specific wide IP is enabled or disabled (the status)
- Disable a specific wide IP for a certain period of time
- Enable a wide IP that is disabled
- View all of the pools for a specific wide IP
- View whether a specific pool is enabled or disabled (the status)
- Disable a specific pool for a certain period of time
- Enable a pool that is disabled
- Get the following information for each virtual server in a wide IP pool:
 - **Enabled** or **disabled** status
 - Availability status: **green** (available), **blue** (unknown), **red** (down), or **yellow** (unavailable)
 - IP address
 - Port
 - Ratio value (for the Ratio load balancing mode)
- View the status of a wide IP, in the context of a data center
- Disable a wide IP, in the context of a data center
- Enable a wide IP, in the context of a data center



C

bigpipe Command Reference

bigpipe commands

This chapter lists the various **bigpipe** commands that are available on the 3-DNS, including syntax requirements and functional descriptions. Table C.1 outlines the conventions used in the command line syntax.

Item in text	Description
\	Continue to the next line without typing a line break.
< >	You enter text for the enclosed item. For example, if the command has <your name> , type in your name.
	Separates alternate options for a command.
[]	Syntax inside the brackets is optional.
...	Indicates that you can type a series of items.

Table C.1 Command line conventions

◆ Note

*You can use both **bigpipe** and **b** to start a **bigpipe** command.*

The following table provides a concise listing of the individual **bigpipe** commands, along with the page reference where you can find the detailed description.

Command	Description	Page
-?	Displays online help for an individual bigpipe command.	C-3
config	Synchronizes the /config/bigip.conf between the two 3-DNS units in a redundant system.	C-4
failover	Sets the 3-DNS as active or standby.	C-5
global	Sets global variable definitions.	C-6
-h and help	Displays online help for bigpipe command syntax.	C-9
interface	Sets options on individual interfaces.	C-10
load	Loads the 3-DNS configuration and resets.	C-11
merge	Loads a saved 3-DNS configuration without resetting the current configuration.	C-12
monitor	Defines a health check monitor.	C-13
reset	Clears the 3-DNS configuration and counter values.	C-14

Command	Description	Page
save	Writes the current configuration to a file.	C-15
self	Assigns a self IP address for a VLAN or interface.	C-16
trunk	Aggregates links to form a trunk.	C-17
unit	Displays the unit number assigned to a particular 3-DNS.	C-18
verify	Parses the command line and checks syntax without executing the specified command.	C-19
version	Displays the bigpipe utility version number.	C-20
vlan	Defines VLANs, VLAN mappings, and VLAN properties.	C-21
vlangroup	Defines VLAN groups.	C-25

-?

```
bigpipe <command> -?
```

For certain commands, displays online help, including complete syntax, description, and other related information. For example, to see online help for the **bigpipe global** command, type:

```
b global -?
```

config

```
b config save <file>
b config install <file>
```

The **bigpipe config** commands archive configuration files for backup purposes (**config save**) and installs saved files (**config install**).

Saving configuration files to an archive

The **config save <file>** command saves all configuration files to a single archive file, **<file>.ucs**, on the local system. By default, **<file>.ucs** is saved to the directory **/user/local/ucs**. An alternate location can be specified by expressing **<file>** as a relative or absolute path. For example:

```
b config save /user/local/config_backup/my_conf
```

This writes the file **my_conf.ucs** to the directory **/user/local/config_backup**.

Installing an archived configuration file

The **config install <file>** command reinstalls the archived configuration files saved as **<file>.ucs** to their working locations on the local system.

If you use command line utilities to set configuration options, be sure to save the current configuration to the relevant files before you use the configuration synchronization feature. (Alternatively, if you want to test the memory version on the standby unit first, use **bigpipe config sync running**.) Use the following **bigpipe** command to save the current configuration:

```
b save
```

◆ **Note**

*A file named **/usr/local/ucs/cs_backup.ucs** is created prior to installing a UCS from a remote machine.*

failover

```
b failover standby | show | init
```

This group of commands affects the fail-over status of the 3-DNS. Note that the **failover** commands are only valid if you have a redundant system.

Run the following command to place a 3-DNS unit in standby mode:

```
b failover standby
```

Show the status of the 3-DNS unit with the following command:

```
b failover show
```

You can use the **bigpipe failover init** command to refresh the parameters of the fail-over mechanism with any new configuration data entered in the BIG/db database.

```
b failover init
```

global

```
b global auto_lasthop enable | disable | show
b global ipforwarding enable | disable
b global open_3dns_ports enable | disable | show
b global open_corba_ports enable | disable | sho
b global open_snmp_ports enable | disable | show
b global open_telnet_port enable | disable
b global open_ftp_ports enable|disable
b global open_ssh_port enable | disable
b global open_rsh_ports enable | disable
b global open_failover_ports enable | disable | show
b global verbose_log_level <level>
b global webadmin_port <port>
b global l2_aging_time <seconds>
```

auto_lasthop

When this variable is enabled, it automatically designates the lasthop router inside IP address as a lasthop route for replies to inbound traffic. If **auto_lasthop** is disabled, the lasthop router inside IP address must be specified as a **lasthop pool**. The default setting is **enable**.

ipforwarding

Enables IP forwarding for the 3-DNS. IP forwarding exposes all of the node IP addresses to the external network, making them routable on that network. Note that this setting is only applicable if you are running the 3-DNS in router mode. The default setting is **disabled**.

open_3dns_ports

This variable is required only when running one or more 3-DNS systems in the network. It does not apply to running the 3-DNS module on a BIG-IP.

open_corba_ports

This variable enables and disables the CORBA ports, which allow administrative CORBA connections. The default setting is **disabled**.

open_snmp_ports

This variable enables and disables the SNMP ports, which allow administrative SNMP connections. The default setting is **disabled**.

open_telnet_port

This variable enables or disables ports for Telnet access. The default setting is **disable**.

The following command sets this variable to open the Telnet port (23) to allow administrative Telnet connections. This is useful for non-crypto 3-DNS systems.

The following command opens the Telnet port:

```
b global open_telnet_port enable
```

The following command closes the Telnet port:

```
b global open_telnet_port disable
```

open_ftp_ports

This variable enables or disables ports for FTP access. The default setting is **disable**.

The following command open the FTP ports (20 and 21) to allow administrative FTP connections, which is useful for non-crypto 3-DNS systems.

```
b global open_ftp_ports enable
```

The following command closes FTP ports:

```
b global open_ftp_ports disable
```

open_ssh_ports

This variable enables or disables ports for SSH access on 3-DNS systems that support encrypted communications. The default setting is **enable**.

The following command opens the SSH port (22) to allow encrypted administrative connections:

```
b global open_ssh_port enable
```

The following command closes the SSH port:

```
b global open_ssh_port disable
```

open_rsh_ports

This variable enables or disables ports for RSH access. You may need to open RSH ports if you are configuring a non-crypto 3-DNS, or if you want a crypto 3-DNS to communicate with non-crypto systems in your network.

The default setting is **disable**.

The following command opens the RSH ports (512, 513, and 514) to allow RSH connections:

```
b global open_rsh_ports enable
```

The following command closes RSH ports:

```
b global open_rsh_ports disable
```

open_failover_ports

This variable enables or disables network failover (failover in a redundant system with no serial cable connection) when a VLAN has port lockdown enabled.

The following command enables network failover:

```
b global open_failover_ports enable
```

The following command disables network failover:

```
b global open_failover_ports disable
```

verbose_log_level

This variable sets logging levels for both TCP and UDP traffic. Each log level is identified by a level number used in place of the **<level>** parameter.

The following command turns on port denial logging for both TCP and UDP traffic. This logs TCP and UDP port denials to the virtual server address and the 3-DNS address.

```
b global verbose_log_level 15
```

The following command turns logging off altogether:

```
b global verbose_log_level 0
```

Setting log levels only for TCP traffic

The following command turns on only TCP port denial logging, which logs TCP port denials to the 3-DNS address.

```
b global verbose_log_level 2
```

The following command turns on virtual TCP port denial logging, which logs TCP port denials to the virtual server address.

```
b global verbose_log_level 8
```

Setting log levels for UDP traffic

The following command turns on only UDP port denial logging, which logs UDP port denials to the 3-DNS address.

```
b global verbose_log_level 1
```

The following command turns on only virtual UDP port denial logging, which logs UDP port denials to the virtual server address.

```
b global verbose_log_level 4
```

webadmin_port

Specifies the port number used for administrative web access. The default port for web administration is port **443**.

-h and -help

```
b [ -h | -help ]
```

Displays the **bigpipe** command syntax or usage text for all current commands.

◆ **Note**

*More detailed man pages are available for some individual **bigpipe** commands. To display detailed online help for the **bigpipe** command, type: **man bigpipe**.*

interface

```
b interface <if_name> media <media_type>|show
b interface <if_name> duplex full|half|auto|show
b interface [<if_name>] show [verbose]
b interface [<if_name>] stats reset
```

Displays names of installed network interface cards and allows you to set properties for each network interface card.

Setting the media type

The media type may be set to the specific media type for the interface card, or it may be set to **auto** for auto detection. If the media type is set to **auto** and the card does not support auto detection, the default type for that interface will be used, for example **100baseTX**.

Setting the duplex mode

Duplex mode may be set to full or half duplex. If the media type does not allow duplex mode to be set, this will be indicated by an onscreen message. If media type is set to **auto**, or if setting duplex mode is not supported, the duplex setting will not be saved to **bigip.conf**.

load

```
b [verify] load [<filename>|-]  
b [-log] load [<filename>|-]
```

Resets all of the 3-DNS settings and then loads, by default, the configuration settings from the **/config/bigip.conf** and **/config/bigip_base.conf** files.

For testing purposes, you can save a test configuration by renaming it to avoid confusion with the boot configuration file. To load a test configuration, use the **load** command with the **<filename>** parameter. For example, if you renamed your configuration file to **/config/bigtest.conf**, the command would be:

```
b load /config/bigtest.conf
```

The command checks the syntax and logic, reporting any errors that would be encountered if the command executed.

You can type **b load -** in place of a file name, to display the configuration on the standard output device.

```
b load -
```

Use the **load** command together with the **verify** command to validate the specified configuration file. For example, to check the syntax of the configuration file **/config/altbigip.conf**, use the following command:

```
b verify load /config/altbigip.conf
```

The **-log** option will cause any error messages to be written to **/var/log/bigip** in addition to the terminal.

merge

```
b [-log] merge [<file_name>]
```

Use the **merge** command to load the 3-DNS configuration from **<file_name>** without resetting the current configuration.

monitor

```
b monitor show [all]
b monitor <name> show
b monitor <name> enable | disable
```

Defines a health monitor. A health monitor is a configuration object that defines how and at what intervals a node is pinged to determine if it is **up** or **down**.

◆ Note

*On a 3-DNS, this **bigpipe** option is applicable only to the default gateway pool, and the default monitor is **icmp**.*

Showing, disabling, and deleting monitors

There are monitor commands for showing, disabling, and deleting monitors.

To show monitors

You can display a selected monitor or all monitors using the **bigpipe monitor show** command:

```
b monitor <name> show
b monitor show all
```

To disable a monitor

All monitors are enabled by default. You can disable a selected monitor, which effectively removes the monitor from service. To disable a monitor, use the **bigpipe monitor <name> disable** command:

```
b monitor <name> disable
```

To re-enable a disabled monitor

Disabled monitors may be re-enabled as follows:

```
b monitor <name> enable
```

reset

```
b reset
```

Use the following syntax to clear the configuration values and counter values from memory:

```
b reset
```

 **WARNING**

Use this command with caution. All network traffic stops when you run this command.

Typically, this command is used on a standby 3-DNS in a redundant system prior to loading a new **/config/bigip.conf** file that contains new service enable and timeout values.

For example, you can execute the following commands on a standby 3-DNS:

```
b reset
b load <filename>
```

This sequence of commands ensures that only the values set in the **<filename>** specified are in use.

save

```
b save [ <filename> | - ]
b base save [ <filename> | - ]
```

The **bigpipe save** and **base save** commands write the current 3-DNS configuration settings from memory to the configuration files named **/config/bigip.conf** and **/config/bigip_base.conf**. (The **/config/bigip.conf** file stores high-level configuration settings, such as pools, virtual servers, NATs, SNATs, and proxies. The **/config/bigip_base.conf** file stores low-level configuration settings, such as VLANs, non-floating self IP addresses, and interface settings.)

You can type **b save <filename>**, or a hyphen character (-) in place of a file name, to display the configuration on the standard output device.

```
b [base] save -
```

If you are testing and integrating 3-DNS systems into a network, you may want to use multiple test configuration files. Use the following syntax to save the current configuration to a file name that you specify:

```
b [base] save <filename>
```

For example, the following command saves the current configuration from memory to an alternate configuration file named **/config/bigip.conf2**.

```
b save /config/bigip.conf2
```

self

```
b self <addr> vlan <vlan_name> [ netmask <ip_mask> ][ broadcast <broadcast_addr>] [unit
  <id>]
b self <addr> floating enable | disable
b self <addr> delete
b self <addr> show
b self show
```

The **self** command defines a self IP address on a 3-DNS. A self IP address is an IP address mapping to a VLAN or VLAN group and their associated interfaces on a 3-DNS. One self IP address is assigned to each interface in the unit as part of the initial system configuration. During the initial system configuration, if you have a redundant system, you also create a floating (shared) self IP address. Additional self IP addresses may be created for health checking, gateway failsafe, routing, or other purposes. These additional self IP addresses are created using the **self** command.

Any number of additional self IP addresses may be added to a VLAN to create aliases. Example:

```
b self 11.11.11.4 vlan external
b self 11.11.11.5 vlan external
b self 11.11.11.6 vlan external
b self 11.11.11.7 vlan external
```

Also, any one self IP address may have **floating** enabled to create a *floating* self IP address that is shared by both units of a 3-DNS redundant system:

```
b self 11.11.11.8 floating enable
```

Assigning a self IP address to a VLAN automatically maps it to the VLAN's interfaces. Since all interfaces must be mapped to one and only one untagged VLAN, assigning a self IP address to an interface not mapped to an untagged VLAN produces an error message.

trunk

```
b trunk <controlling_if> define <if_list>
b trunk [<controlling_if>] show [verbose]
b trunk [<controlling_if>] stats reset
```

The **trunk** command aggregates links (individual physical interfaces) to form a trunk. Link aggregation increases the bandwidth of the individual NICs in an additive manner. Thus, four fast Ethernet links, if aggregated, create a single 400 Mb/s link. The other advantage of link aggregation is link failover. If one link in a trunk goes down, traffic is simply redistributed over the remaining links.

A trunk must have a controlling link, and acquires all the attributes of that controlling link from Layer 2 and above. Thus, the trunk automatically acquires the VLAN membership of the controlling link, but does not acquire its media type and speed. Outbound packets to the controlling link are load balanced across all of the known-good links in the trunk. Inbound packets from any link in the trunk are treated as if they came from the controlling link.

A maximum of eight links may be aggregated. For optimal performance, links should be aggregated in powers of two. Thus ideally, you will aggregate two, four, or eight links. Gigabit and fast Ethernet links cannot be placed in the same trunk.

Creating a trunk

To create a trunk, use the following syntax:

```
b trunk <controlling_if> define <if_list>
```

Interfaces are specified using the **s.p** convention, where **s** is slot number and **p** is port number. An **<if_list>** is one or more such interfaces, with multiple interfaces separated by spaces or commas. A range may be specified as follows:

```
2.1-2.7
```

For more information on interface naming, refer to the *3-DNS Administrator Guide*, Chapter 5, *Configuring the Base Network*.

unit

```
b unit [show]
b unit peer [show]
```

The unit number on a 3-DNS designates which virtual servers use a particular unit in an active-active redundant configuration. You can use the **bigpipe unit** command to display the unit number assigned to a particular 3-DNS. For example, to display the unit number of the unit you are on, type the following command:

```
b unit show
```

To display the unit number of the other 3-DNS in a redundant system, type in the following command:

```
b unit peer show
```

◆ **Note**

If you use this command on a redundant system in active/standby mode, the active unit shows as unit 1 and 2, and the standby unit has no unit numbers.

◆ **Tip**

*The **bigpipe unit peer show** command is the best way to determine whether the respective state mirroring mechanisms are connected.*

verify

```
b [log] verify <command...>
verify load [<filename>|-]
```

Parses the command line and checks syntax without executing the specified command. This distinguishes between valid and invalid commands

Use the **verify** command followed by a command that you want to validate. For example, to verify that the vlans **external1** and **external2** have been added to the VLAN group **bridge**, type the following command:

```
b verify vlangroup bridge vlans add external1 external2
```

The command checks the syntax and logic, and reports any errors that would be encountered if the command executed.

Use the **verify** command together with the **load <filename>** command to validate the specified configuration file. For example, to check the syntax of the configuration file **/config/altbigip.conf**, use the following command:

```
b verify load /config/altbigip.conf
```

version

```
b version
```

Displays the version of the 3-DNS operating system and the features that are enabled.

For example, for a 3-DNS HA, the **bigpipe version** command displays the output shown in Figure C.1.

```
Product Code:
3-DNS HA

Enabled Features:
SSL Gateway                Gateway Failsafe
Static Load Balancing      Snat
Nat                        Pools
Akamaizer                  Full Proxy
Late Binding               HTTP Rules
Mirroring                  Failover
Node HA                    Dynamic Load Balancing
Destination Address Affinity Cookie Persistence
SSL Persistence            Simple Persistence
EAV                        ECV SSL
ECV                        ECV Transparent
Health Check               Filter
```

Figure C.1 *The version output display*

vlan

```

b vlan <vlan_name>
b vlan <name> rename <new_name>
b vlan <vlan_name> delete
b vlan <vlan_name> tag <tag_number>
b vlan <vlan_name> interfaces add [tagged] <if_list>
b vlan <vlan_name> interfaces delete <if_list>
b vlan <vlan_name> interfaces delete all
b vlan <vlan_name> interfaces show
b vlan <vlan_name> port_lockdown enable | disable
b vlan <vlangroup_name> proxy_forward enable | disable
b vlan <vlan_name> failsafe arm|disarm|show
b vlan <vlan_name> timeout <seconds>|show
b vlan show
b vlan <vlan_name> show
b vlan <vlan_name> interfaces show
b vlan <vlan_name> rename <new_vlan_name>
b vlan <if_name> mac_masq <mac_addr> | show
b vlan <if_name> mac_masq 0:0:0:0:0

```

The **vlan** command defines VLANs, VLAN mappings, and VLAN properties. By default, each interface on a 3-DNS is an untagged member of an interface-group VLAN. The lowest-numbered interface is assigned to the **external** VLAN, the interface on the main board is assigned to the **admin** VLAN, and all other interfaces are assigned to the **internal** VLAN.

Using the **vlan** command, you can create tagged and untagged VLANs, make and change assignments of VLANs to interfaces, and configure a range of VLAN attributes. This includes enabling/disabling of port lockdown, arming and disarming failsafe, and setting the failure timeout. Table C.2 shows the VLAN configuration options.

Attributes	Description
Default VLAN configuration	The Setup utility provides a default VLAN configuration. On a typical unit with two interfaces, you create an internal and external VLAN.
VLAN	Create, rename, or delete a VLAN. Typically, one VLAN is assigned to one interface.
Tag VLANs	You can tag VLANs and add multiple tagged VLANs to a single interface.
VLAN security	You can set port lockdown by VLAN.

Table C.2 VLAN configuration options

Attributes	Description
Set fail-safe timeouts	You can set a failsafe timeout on a VLAN. You can use a failsafe timeout to trigger fail-over in a redundant system.
Self IP addresses	You can set self IP addresses for VLANs.
MAC masquerade	You can use this attribute to set up a media access control (MAC) address that is shared by redundant units. This allows you to use the 3-DNS units in a topology with secure hubs.

Table C.2 VLAN configuration options

Creating and assigning a VLAN

To create a VLAN, use the following syntax:

```
b vlan <name>
```

<name> is typically symbolic, as in:

```
b vlan vlan5
```

Typically you define a VLAN and specify the interfaces on the VLAN in the same command:

```
b vlan vlan5 interfaces add [tagged] <if_list>
```

Tagged VLANs

A new tagged VLAN is created using the **bigpipe vlan tag** command, specifying a tag number. For example:

```
b vlan my_vlan tag 1209
```

A tagged VLAN is mapped to an interface or interfaces (or an untagged VLAN is tagged and mapped an interface or interfaces) using the **tagged** flag. For example:

```
b vlan external interfaces add tagged 4.1 5.1 5.2
```

The effect of the command is to place a tag on interfaces **4.1** and **5.1**, which in turn makes **external** a tagged VLAN. (However, it remains an untagged VLAN for interfaces which are part of it but not tagged.)

An interface can have more than one tag; it can be a member of more than one tagged VLAN.

```
b vlan external interfaces add tagged 4.1
```

```
b vlan internal interfaces add tagged 4.1
```

```
b vlan admin interfaces add tagged 4.1
```

This permits tagged VLANs to form a VLAN trunk on a single interface.

Enabling and disabling port lockdown

You can lock down a VLAN to prevent direct connection to the 3-DNS through that VLAN using the following command:

```
b vlan <vlan_name> port_lockdown enable
```

Note that you do not want to enable port lockdown on a 3-DNS on which you are only using a single VLAN.

Setting the fail-over timeout and arming the fail-safe

For redundant 3-DNS systems, failover (activation of the inactive system) occurs when loss of traffic is detected on a VLAN and traffic is not restored during the failover timeout period for that VLAN. You can enable a fail-safe mechanism to attempt to generate traffic when half the timeout has elapsed. If the attempt is successful, the failover is stopped.

Using the **vlan** command, you may set the timeout period and also arm or disarm the fail-safe.

To set the timeout, type the following command:

```
b vlan <vlan_name> timeout <timeout_in_seconds>
```

To arm the failsafe, use this command:

```
b vlan <vlan_name> failsafe arm
```

To disarm the failsafe, use this syntax:

```
b vlan <vlan_name> failsafe disarm
```

Setting the MAC masquerade address

Sharing the MAC masquerade address makes it possible to use 3-DNS systems in a network topology using secure hubs. The MAC address for a VLAN is the first interface to which the VLAN is mapped. You can view the VLAN-to-interface mapping using the following command:

```
b vlan show
```

You can view the media access control (MAC) address on a given unit using the following command:

```
b interface show
```

Use the following syntax to set the MAC masquerade address that will be shared by both 3-DNS units in the redundant system.

```
b vlan <vlan_name> mac_masq <MAC_addr>
```

WARNING

*You must specify a default route before using the **mac_masq** command. You specify the default route in the **/etc/hosts** and **/etc/netstart** files.*

Find the MAC address on both the active and standby units and choose one that is similar but unique. A safe technique for choosing the shared MAC address follows:

Suppose you want to set up **mac_masq** on the external interfaces. Using the **bigpipe interface show** command on the active and standby units, you note that their MAC addresses are:

Active: 3.1 = 0:0:0:ac:4c:a2

Standby: 3.1 = 0:0:0:ad:4d:f3

In order to avoid packet collisions, you now must choose a unique MAC address. The safest way to do this is to select one of the addresses and logically **OR** the first byte with **0x40**. This makes the MAC address a locally administered MAC address.

In this example, either **40:0:0:ac:4c:a2** or **40:0:0:ad:4d:f3** would be a suitable shared MAC address to use on both 3-DNS units in the redundant system.

The shared MAC address is used only when the 3-DNS is in active mode. When the 3-DNS is in standby mode, the original MAC address of the network card is used.

If you do not configure **mac_masq** on startup, or when transitioning from standby mode to active mode, the 3-DNS sends gratuitous ARP requests to notify the default router and other machines on the local Ethernet segment that its MAC address has changed. See RFC 826 for more details on ARP.

◆ **Note**

You can use the same technique to configure a shared MAC address for each interface.

vlangroup

```
b vlangroup <vlangroup_name> { vlans add <vlan_list> }  
b vlan <vlangroup_name> proxy_forward enable | disable  
b vlangroup <vlangroup_name> delete
```

The **vlangroup** command defines a VLAN group, which is a grouping of two or more VLANs belonging to the same IP network for the purpose of allowing L2 packet forwarding between those VLANs.

The VLANs between which the packets are to be passed must be on the same IP network, and they must be grouped using the **vlangroup** command. For example:

```
b vlangroup network11 { vlans add internal external }
```

A self IP address must be assigned to the VLAN group using the following command:

```
b self <ip_addr> vlan network11
```

L2 forwarding must be enabled for the VLAN group using the **vlan proxy_forward** attribute. This attribute is enabled by default when the VLAN group is enabled.



Glossary

3-DNS Distributed Traffic Controller

The 3-DNS Distributed Traffic Controller is a wide area load distribution solution that intelligently allocates Internet and intranet service requests across geographically distributed network servers. The 3-DNS Distributed Traffic Controller is also called the 3-DNS.

3-DNS Maintenance menu

The 3-DNS Maintenance menu is a command line utility that you use to configure the 3-DNS.

3-DNS web server

The 3-DNS web server is a standard web server that hosts the Configuration utility on the 3-DNS.

A record

The **A** record is the ADDRESS resource record that a 3-DNS returns to a local DNS server in response to a name resolution request. The **A** record contains a variety of information, including one or more IP addresses that resolve to the requested domain name.

access control list (ACL)

An access control list is a list of local DNS server IP addresses that are excluded from path probing or hops queries.

active unit

In a redundant system, an active unit is a system that currently load balances name resolution requests. If the active unit in the redundant system fails, the standby unit assumes control and begins to load balance requests.

alternate method

The alternate method specifies the load balancing mode that the 3-DNS uses to pick a virtual server if the preferred method fails. See also *fallback method*, *preferred method*.

big3d agent

The **big3d** agent is a monitoring agent that collects metrics information about server performance and network paths between a data center and a specific local DNS server. The 3-DNS uses the information collected by the **big3d** agent for dynamic load balancing.

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>.

CDN switching

CDN switching is the functionality of the 3-DNS that allows a user to redirect traffic to a third-party network, or transparently switch traffic to a CDN. The two features of the 3-DNS that make CDN switching possible are geographic redirection and the pool type CDN.

CNAME record

A canonical name (CNAME) record acts as an alias to another domain name. A canonical name and its alias can belong to different zones, so the **CNAME** record must always be entered as a fully qualified domain name. **CNAME** records are useful for setting up logical names for network services so that they can be easily relocated to different physical hosts.

completion rate

The completion rate is the percentage of packets that a server successfully returns during a given session.

Completion Rate mode

The Completion Rate mode is a dynamic load balancing mode that distributes connections based on which network path drops the fewest packets, or allows the fewest number of packets to time out.

Configuration utility

The Configuration utility is the browser-based application that you use to configure the 3-DNS.

content delivery network (CDN)

A content delivery network (CDN) is an architecture of Web-based network components that helps dramatically reduce the wide-area network latency between a client and the content they wish to access. A CDN includes some or all of the following network components: wide-area traffic managers, Internet service providers, content server clusters, caches, and origin content providers.

data center

A data center is a physical location that houses one or more 3-DNS systems, BIG-IP systems, EDGE-FX Caches, GLOBAL-SITE systems, or host machines.

data center server

A data center server is any server recognized in the 3-DNS configuration. A data center server can be any of the following: a 3-DNS, a BIG-IP, an EDGE-FX Cache, a GLOBAL-SITE, or a host.

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.f5.com/index.html>, the domain name is **f5.com**.

dynamic load balancing modes

Dynamic load balancing modes base the distribution of name resolution requests to virtual servers on live data, such as current server performance and current connection load.

dynamic site content

Dynamic site content is a type of site content that is automatically generated each time a user accesses the site. Examples are current stock quotes or weather satellite images.

Extended Content Verification (ECV)

On the 3-DNS, ECV is a service monitor that checks the availability of actual content, (such as a file or an image) on a server, rather than just checking the availability of a port or service, such as HTTP on port 80.

external interface

An external interface is the network interface that can be accessed across a wide-area network (WAN). See also *internal interface*.

fail-over

Fail-over is the process whereby a standby unit in a redundant system takes over when a software failure or hardware failure is detected on the active unit.

fail-over cable

The fail-over cable is the cable that directly connects the two system units in a hardware-based redundant system.

fallback method

The fallback method is the third method in a load balancing hierarchy that the 3-DNS uses to load balance a resolution request. The 3-DNS uses the fallback method only when the load balancing modes specified for the preferred and alternate methods fail. Unlike the preferred method and the alternate method, the fallback method uses neither server nor virtual server availability for load balancing calculations. See also *preferred method*, *alternate method*.

FDDI (Fiber Distributed Data Interface)

FDDI is a multi-mode protocol for transmitting data on optical-fiber cables at speeds up to 100 Mbps.

Global Availability mode

Global Availability is a static load balancing mode that bases connection distribution on a particular server order, always sending a connection to the first available server in the list. This mode differs from Round Robin mode in that it searches for an available server always starting with the first server in the list, while Round Robin mode searches for an available server starting with the next server in the list (with respect to the server selected for the previous connection request).

hops factory

A hops factory is a type of factory run by the **big3d** agent that collects hops data about network paths.

host

A host is a network server that manages one or more virtual servers that the 3-DNS uses for load balancing.

ICMP (Internet Control Message Protocol)

ICMP is an Internet communications protocol used to determine information about routes to destination addresses, such as virtual servers managed by BIG-IP systems and 3-DNS systems.

internal interface

An internal interface is a network interface that can be accessed from a local-area network (LAN). See also *external interface*.

iQuery

The iQuery protocol is used to exchange information between 3-DNS systems, BIG-IP systems, EDGE-FX Caches, and GLOBAL-SITE systems. The iQuery protocol is officially registered with IANA for port 4353, and works on UDP and TCP connections.

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.

Least Connections mode

The Least Connections mode is a dynamic load balancing mode that bases connection distribution on which server currently manages the fewest open connections.

load balancing methods

Load balancing methods are the settings that specify the hierarchical order in which the 3-DNS uses three load balancing modes. The preferred method specifies the first load balancing mode that the 3-DNS tries, the alternate

method specifies the next load balancing mode to try if the preferred method fails, and the fallback method specifies the last load balancing mode to use if both the preferred and the alternate methods fail.

load balancing mode

A load balancing mode is the way in which the 3-DNS determines how to distribute connections across an array.

local DNS

A local DNS is a server that makes name resolution requests on behalf of a client. With respect to the 3-DNS, local DNS servers are the source of name resolution requests. Local DNS is also referred to as LDNS.

metrics information

Metrics information is the data that is typically collected about the paths between BIG-IP systems, EDGE-FX Caches or GLOBAL-SITE systems, and local DNS servers. Metrics information is also collected about the performance and availability of virtual servers. Metrics information is used for load balancing, and it can include statistics such as round trip time, packet rate, and packet loss.

MindTerm SSH

MindTerm SSH is the third-party application on 3-DNS systems that uses SSH for secure remote communications. SSH encrypts all network traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks. SSH also provides secure tunneling capabilities and a variety of authentication methods.

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.

name server

A name server is a server that maintains a DNS database, and resolves domain name requests to IP addresses using that database.

named

The **named** daemon manages domain name server software.

NameSurfer

NameSurfer is the third-party application on 3-DNS systems that automatically manages DNS zone files, synchronizing them with the configuration on the system. NameSurfer automatically updates any configuration changes that you make using the Configuration utility. NameSurfer also provides a graphical user interface for DNS zone file management.

Network Time Protocol (NTP)

Network Time Protocol functions over the Internet to synchronize system clocks to Universal Coordinated Time. NTP provides a mechanism to set and maintain clock synchronization within milliseconds.

NS record

A name server (NS) record is used to define a set of authoritative name servers for a DNS zone. A name server is considered authoritative for some given zone when it has a complete set of data for the zone, allowing it to answer queries about the zone on its own, without needing to consult another name server.

packet rate

The packet rate is the number of data packets per second processed by a server.

Packet Rate mode

The Packet Rate mode is a dynamic load balancing mode that distributes connections based on which available server currently processes the fewest packets per second.

path

A path is a logical network route between a data center server and a local DNS server.

path probing

Path probing is the collection of metrics data, such as round trip time and packet rate, for a given path between a requesting LDNS server and a data center server.

persistence

On a 3-DNS, persistence is a series of related requests received from the same local DNS server for the same wide IP name. When persistence is turned on, a 3-DNS sends all requests from a particular local DNS server for a specific wide IP to the same virtual server, instead of load balancing the requests.

picks

Picks represent the number of times a particular virtual server is selected to receive a load balanced connection.

pool

A pool is a group of virtual servers managed by a BIG-IP, an EDGE-FX Cache, or a host. The 3-DNS load balances among pools (using the Pool LB Mode), as well as among individual virtual servers.

pool ratio

A pool ratio is a ratio weight applied to pools in a wide IP. If the Pool LB mode is set to Ratio, the 3-DNS uses each pool for load balancing in proportion to the weight defined for the pool.

preferred method

The preferred method specifies the first load balancing mode that the 3-DNS uses to load balance a resolution request. See also *alternate method*, *fallback method*.

principal 3-DNS

A 3-DNS that initiates metrics collection by the **big3d** agents and distributes the metrics to other members of a sync group. See also *receiver 3-DNS*.

probe protocol

The probe protocol is the specific protocol used to probe a given path and collect metrics information for the path. The probe protocols available on the 3-DNS are: ICMP, DNS_REV, DNS_DOT, UDP, and TCP. The probe protocols that are available change based on the data center server type.

prober

A prober is a specific thread of the **big3d** agent that is used for path probing of a given set of paths.

prober factory

A prober factory is a utility that collects metrics data, such as round trip time and packet rate, for a given path between a requesting LDNS and a data center server. Prober factories are managed by the **big3d** agent, which reports the path probing metrics to the 3-DNS. Prober factories can run on BIG-IP systems, EDGE-FX Caches, and GLOBAL-SITE systems.

production rule

A production rule, on the 3-DNS, can change system behavior under specific operating conditions. For example, a production rule can switch load balancing modes or can reroute network traffic to a specific set of servers. Production rules are based on triggers such as time of day or current network traffic load.

QOS equation

The QOS equation is the equation on which the Quality of Service load balancing mode is based. The equation calculates a score for a given path between a data center server and a local DNS server. The Quality of Service mode distributes connections based on the best path score for an available data center server. You can apply weights to the factors in the equation, such as round trip time and completion rate.

Quality of Service load balancing mode

The Quality of Service load balancing mode is a dynamic load balancing mode that bases connection distribution on a configurable combination of the packet rate, completion rate, round trip time, hops, virtual server capacity, kilobytes per second, and topology information.

ratio

A ratio is the parameter in a virtual server statement that assigns a weight to the virtual server for load balancing purposes.

Ratio mode

The Ratio load balancing mode is a static load balancing mode that distributes connections across an pool of virtual servers in proportion to the ratio weight assigned to each individual virtual server.

receiver 3-DNS

A receiver 3-DNS is a system, in a sync group, that receives metrics data that are broadcast from **big3d** agents, but does not initiate metrics collection. See also *principal 3-DNS*.

redundant system

A redundant system is a pair of systems that are configured for fail-over. In a redundant system, one system runs as the active unit and the other system runs as the standby unit. If the active unit fails, the standby unit takes over and manages resolution requests.

remote administrative IP address

A remote administrative IP address is an IP address from which a system allows shell connections, such as SSH, RSH, or Telnet.

resolver

The resolver is the client part of the Domain Name System. The resolver translates a program's request for host name information into a query to a name server, and translates the response into an answer to the program's request. See also *name server*.

resource record

resource record is a record in a DNS database that stores data associated with domain names. A resource record typically includes a domain name, a TTL, a record type, and data specific to that record type. See also *A record*, *CNAME record*, *NS record*.

reverse domains

A type of DNS resolution request that matches a given IP address to a domain name. The more common type of DNS resolution request starts with a given domain name and matches that to an IP address.

root name server

A root name server is a master DNS server that maintains a complete DNS database. There are approximately 13 root name servers in the world that manage the DNS database for the World Wide Web.

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.

round trip time (RTT)

Round trip time is the calculation of the time (in microseconds) that a local DNS server takes to respond to a ping issued by the **big3d** agent running on a data center server. The 3-DNS takes RTT values into account when it uses dynamic load balancing modes.

Round Trip Time mode

Round Trip Time is a dynamic load balancing mode that bases connection distribution on which virtual server has the fastest measured round trip time between the data center server and the local DNS server.

secondary DNS

The secondary DNS is a name server that retrieves DNS data from the name server that is authoritative for the DNS zone.

Setup utility

The Setup utility is a utility that takes you through the initial system configuration process. The Setup utility runs automatically when you turn on a system for the first time.

site content

Site content is data (including text, images, audio, and video feeds) that is accessible to clients who connect to a given site. See also *dynamic site content*, *static site content*.

SNMP (Simple Network Management Protocol)

SNMP is the Internet standard protocol, defined in STD 15, RFC 1157, that was developed to manage nodes on an IP network.

sod (switch over daemon)

The **sod** daemon controls the fail-over process in a redundant system.

SSH

SSH is a protocol for secure remote login and other secure network services over a non-secure network.

standby unit

A standby unit is a system in a redundant system that is always prepared to become the active unit if the active unit fails.

static load balancing modes

Static load balancing modes base the distribution of name resolution requests to virtual servers on a pre-defined list of criteria and server and virtual server availability; they do not take current server performance or current connection load into account.

static site content

Static site content is a type of site content that is stored in HTML pages, and changes only when an administrator edits the HTML document itself.

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.

sub-statement

A sub-statement is a logical section within a statement that defines a particular element in the statement. A sub-statement begins with the sub-statement name followed by an open brace ({) and ends with a closed brace (}). Everything between those braces is part of the sub-statement. Sub-statements typically define a group of related variables, such as the calculation coefficients used in Quality of Service load balancing.

sync group

A sync group is a group of 3-DNS systems that share system configurations and path metrics for data center servers and virtual servers. Sync groups have one principal 3-DNS, and may contain one or more receiver systems. The receiver systems obtain their configuration information from the principal system. See also *principal 3-DNS*, *receiver 3-DNS*.

time tolerance value

The time tolerance value is the number of seconds that one system's clock is allowed to differ in comparison to another system's clock, without the two clocks being considered out of sync.

Topology mode

The Topology mode is a static load balancing mode that bases the distribution of name resolution requests on the weighted scores for topology records. Topology records are used by the Topology load balancing mode to redirect DNS queries to the closest virtual server, geographically, based on location information derived from the DNS query message.

topology record

A topology record specifies a score for a local DNS server location endpoint and a virtual server location endpoint.

topology score

The topology score is the weight assigned to a topology record when the 3-DNS is filtering the topology records to find the best virtual server match for a DNS query.

topology statement

A topology statement is a collection of topology records.

traceroute

Traceroute is the utility that the hops factory uses to calculate the total number of network hops between a local DNS server and a specific data center.

TTL (Time to Live)

The TTL is the number of seconds for which a specific DNS record or metric is considered to be valid. When a TTL expires, the server usually must refresh the information before using it again.

unavailable

The **unavailable** is a status used for data center servers and virtual servers. When a data center server or virtual server is **unavailable**, the 3-DNS does not use it for load balancing.

unknown

The **unknown** status is used for data center servers and virtual servers. When a data center server or virtual server is new to the 3-DNS and does not yet have metrics information, the 3-DNS marks its status as **unknown**. The 3-DNS can use unknown servers for load balancing, but if the load balancing mode is dynamic, the 3-DNS uses default metrics information for the unknown server until it receives live metrics data.

up

The **up** status is used for data center servers and virtual servers. When a data center server or virtual server is **up**, the data center server or virtual server is available to respond to name resolution requests.

virtual server

A virtual server is a specific combination of a virtual IP address and virtual port, and is associated with a content site that is managed by a BIG-IP, EDGE-FX Cache, or host server.

watchdog timer card

The watchdog timer card is a hardware device that monitors the 3-DNS for hardware failure.

wide IP

A wide IP is a collection of one or more domain names that maps to one or more groups of virtual servers managed either by BIG-IP systems, EDGE-FX Caches, or by host servers. The 3-DNS load balances name resolution requests across the virtual servers that are defined in the wide IP that is associated with the requested domain name.

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**).

WKS record

A WKS record is a DNS resource record that describes the services usually provided by a particular protocol on a specific port.

zone

In DNS terms, a zone is a subset of DNS records for one or more domains.

zone file

In DNS terms, a zone file is a database set of domains with one or many domain names, designated mail servers, a list of other name servers that can answer resolution requests, and a set of zone attributes, which are contained in an SOA record.



Index

/config/bigip.conf C-15
 /etc/hosts.allow file 12-2
 /etc/snmptrap.conf file 12-5
 /etc/syslog.conf file 12-5
 -? command C-3

3-DNS Maintenance menu
 about 2-1
 and scripts 11-1
 using 2-2
 working with commands 2-1

3-DNS scripts 11-1

3-DNS web server
 configuring 11-1
 configuring passwords 11-1
 configuring users 11-1

3dns_add script 11-1
 3dns_admin_start 11-1
 3dns_dump script 11-1
 3dns_web_config script 11-1
 3dns_web_passwd script 11-1
 3dnsmaint command. See 3-DNS Maintenance menu
 3dnsmaint script 11-2

3dpipe command
 datacenter B-3
 server type B-5
 stats B-6
 syncgroup B-7
 version B-8
 virtual B-9
 wideip B-10

3dpipe commands
 displaying help B-4

3dprint script 2-2, 11-2
 3dscrip, managing production rules 10-5
 3ndc script 11-3

A

A record, about 5-2
 access control lists
 about 3-1
 defining 3-1
 examples 3-2
 syntax A-40

ACL
 see access control list 3-1

actions supported by production rules 10-9
 additional 3-DNS systems, adding to network 11-1

B

big3d agent
 about 4-1
 and dynamic load balancing 8-5
 and firewalls 4-9
 collecting path data 4-11
 configuring 4-1
 default settings 4-1
 distributing in the network 4-6
 installing on BIG-IP 4-1, 4-2
 installing on EDGE-FX Caches 4-2
 installing on GLOBAL-SITE 4-2
 restarting 2-5, 11-4
 stopping 11-4
 updating 2-4
 updating BIG-IP 11-3
 updating EDGE-FX Caches 11-3
 updating GLOBAL-SITE 11-3
 using factories 4-3
 viewing version numbers 11-4
 working with 2-4

big3d_install script 2-5, 11-3
 big3d_restart script 11-4
 big3d_restart script, 2-5
 big3d_version script 11-4

BIG-IP
 collecting metrics 12-8
 installing big3d agent 4-1, 4-2
 updating big3d agent 2-4, 11-3
 verifying big3d versions 2-4
 viewing big3d version 11-4

bigip.open_3dns_lockdown_ports variable C-6
 bigip.open_ftp_ports variable C-7
 bigip.open_rsh_ports variable C-7
 bigip.open_ssh_port variable C-7
 bigip.open_telnet_port variable C-7
 bigip.verbose_log_level variable C-8

bigpipe commands C-1
 -? C-3
 config C-4
 displaying help C-9
 failover C-5
 global C-6
 interface C-10
 reset C-14
 save C-15
 self C-16
 trunk C-17
 unit C-18
 virtual C-21
 vlan C-21

BIND resources 8-22
 buffer size A-16

C

Change/Add Users for 3-DNS Configuration Utility
command 2-6, 11-1
Check big3d versions command 2-4
Check remote versions of big3d command 11-4
checkpoint file 11-5
checktrap.pl script 12-5
 configuring 12-6
CNAME record, about 5-2
command line
 conventions 1-3, B-1, C-1
 viewing statistics 11-2
commands
 3dpipe B-1
 bigpipe C-1
 Change/Add Users for 3-DNS Configuration Utility
 2-6
 Check remote versions of big3d 2-4
 Configure Connection to NTP Time Server 2-7
 Configure NameSurfer 2-7
 Configure SSH communication with remote devices
 2-5
 Dump 3dnssd Statistics 2-2
 Edit 3-DNS Configuration 2-2
 Edit big3d matrix 2-4
 Edit BIND Configuration 2-2
 Generate and Copy iQuery Encryption key 2-5
 Install and Start big3d 2-5
 Reconfigure 3-DNS Configuration Utility 2-6
 Restart 3-DNS Configuration Utility 2-6
 Restart big3d command 2-5
 Restart syncd 2-7
 Stop syncd 2-7
comments
 in the configuration file A-42
 syntax A-42
 syntax in wideip.conf A-42
config_ssh script 2-5, 11-4
configuration file
 about A-1
configuration file syntax
 for 3-DNS A-18
 for BIG-IP A-20
 for comments A-42
 for datacenter statement A-29
 for EDGE-FX Cache A-22
 for GLOBAL-SITE A-23
 for hosts A-23
 for topology statement A-39
 for wideip statement A-32
configuration settings C-15
Configuration utility
 adding users 2-6
 and custom production rules 10-5
 changing passwords 2-6
 configuring 3-DNS web server 2-6

 setting up production rules 10-1
 updating users 2-6
configurations
 clearing memory C-14
 synchronizing C-4
Configure Connection to NTP Time Server command 2-7
Configure NameSurfer command 2-7
Configure SSH communication with remote devices
command 2-5, 11-4
connections
 FTP C-7
 Telnet C-7
CORBA ports C-6
custom production rules
 and local DNS servers 10-11
 and the Configuration utility 10-5
examples 10-10

D

data center servers
 collecting metrics 12-8
data centers
 analyzing performance 7-5
Dump 3dnssd Statistics command 2-2, 11-2
dynamic load balancing
 about 8-5
 and big3d agents 8-5
dynamic load balancing modes
 and Internet Weather Map 7-3
 and path information A-10
 types 8-5
dynamic ratio
 and Quality of Service mode 8-8

E

e-commerce site
 configuring wide IPs 8-20
ECV 6-1
 search string 6-2
ECV service monitors 6-1
ECV sub-statements A-35
EDGE-FX Cache
 collecting metrics 12-8
 installing big3d agent 4-2
 updating big3d agent 2-4, 11-3
 verifying big3d versions 2-4
 viewing big3d version 11-4
Edit 3-DNS Configuration command 2-2, 11-4
Edit big3d matrix command 2-4
Edit BIND Configuration command 2-2
edit_lock script 11-4
edit_wideip script 11-4

encryption 11-5
 and SSH communications 2-5
 event-based triggers
 defining 10-4
 every statement
 guidelines 10-9
 in production rules 10-9
 examples of syntax, see syntax example.
 Extended Content Verification. See ECV

F

F5makekey script 11-5
 factories
 default settings 4-3
 modifying settings 4-3
 factories, types of
 ECV 4-3
 hops 4-3
 probing 4-3
 SNMP 4-3
 failover command C-5
 file monitors 6-1
 firewalls
 and iQuery 4-9
 configuring for 4-9

FTP

configuring wide IPs 8-20
 open FTP ports C-7

G

Generate and Copy iQuery Encryption Key command
 2-5, 11-5
 global command C-6
 global production rules 10-2
 global timers
 configuring 8-18
 global variables
 configuring load balancing 8-15
 configuring timers 8-19
 configuring TTL 8-19
 globals statement
 about A-5
 load balancing variables 8-17
 GLOBAL-SITE
 installing big3d agent 4-2
 updating big3d agent 2-4, 11-3
 verifying big3d versions 2-4
 viewing big3d version 11-4

H

-h command
 for 3dpipe B-4
 for bigpipe C-9
 hacker detection 10-12

health monitor, see service monitors.
 -help command
 for 3dpipe B-4
 for bigpipe C-9
 help, finding 1-3
 high availability 8-20
 host servers
 collecting metrics 12-8
 configuring SNMP 12-7, 12-8

I

if statement
 guidelines 10-6
 in production rules 10-6
 include files A-2
 Install and Start big3d command 2-5, 4-2
 install_key script 11-5
 interface command C-10
 Internet Weather Map
 about 7-3
 interpreting the data 7-5
 iQuery
 and firewalls 4-9
 configuring for firewalls 4-9
 encrypting 11-5
 iQuery encryption key 4-7
 iQuery key
 distributing 11-5
 iQuery port A-17
 iQuery protocol
 and NTP 2-7

L

last resort pool
 about 8-14
 configuring 8-14, 8-15
 configuring an overflow network 8-15
 lasthop router C-6
 LDNS
 load balancing 10-11
 LDNS persistence
 path information A-10
 LDNS round robin
 about 8-14
 load balancing
 according to LDNS 10-11
 according to time of day 10-10
 and persistence A-10
 configuring 8-9
 configuring at the global level 8-9
 configuring at the wide IP level 8-10
 configuring global variables 8-15
 using production rules 10-10

load balancing modes

- Completion Rate 8-6
- Global Availability 8-3
- Hops 8-6
- Kilobytes/Second 8-6
- Least Connections 8-7
- None 8-3
- Packet Rate 8-7
- Quality of Service 8-8
- Random 8-3
- Ratio 8-4
- Return to DNS 8-4
- Round Robin 8-4
- Round Trip Times (RTT) 8-7
- Static Persist 8-5
- VS Capacity 8-9

load balancing sub-statement A-15

logging C-8

M

MAC addresses C-23

management tool

- production rules 10-1

manual configurations

- troubleshooting 8-22
- troubleshooting syntax errors 8-22
- understanding error messages 8-22
- verifying wideip.conf syntax 8-22, A-1

Map, Network 9-1

media access control. See MAC addresses

memory allocation A-13, A-16

metrics

- and probing A-17
- collecting path information A-12
- setting durations A-11
- types of 12-8
- updating A-10

metrics collection

- about 12-8
- about TTL and timers 8-17
- setting TTL and timer values 8-17

monitors, file 6-1

multiple services

- configuring ports 8-20

MX record 5-3

N

NameSurfer

- and 3-DNS Maintenance menu 2-1
- configuring 2-7

Network Map 9-1

- and objects 9-2
- configuring the network 9-2
- viewing 9-2

network time protocol. See NTP

network traffic

- controlling 10-1

network, viewing layout 9-1

NS record 5-3

NTP

- and iQuery protocol 2-7
- synchronizing 3-DNS systems 2-7
- synchronizing time 2-7

O

overflow network

- and last resort pool 8-15

P

path information

- for persistence A-10
- using probing A-17

path metrics

- collecting 4-5

performance

- analyzing data centers 7-5
- evaluating big3d agent settings 4-5

periodic task intervals A-10

persistence A-10

pools 8-10, A-36

ports

- RSH C-6, C-7

prober sub-statement A-15

probing

- and SNMP 12-7
- configuring A-17
- server types 12-8
- types of metrics 12-8

probing exclusion lists

- see access control lists A-40

production rules 10-1

- according to LDNS 10-11
- according to time of day 10-10
- adding 10-2
- applying a combined date and time variable 10-3
- applying a date variable 10-3
- applying day of the week variable 10-3
- applying time of day variable 10-3
- choosing rule types 10-2
- configuring custom 10-5
- configuring in wideip.conf file 10-5
- defining custom 10-5
- defining global 10-2
- defining triggers 10-2, 10-4
- defining wide IP 10-2
- deleting 10-2
- detecting hackers 10-12
- examples of custom 10-10
- executing 10-5
- getting help 10-5

- inserting in wideip.conf file 10-5
 - managing with 3dscrip utility 10-5
 - managing with Configuration utility 10-2
 - types of actions 10-9
 - understanding 3dscrip guidelines 10-5
 - using Configuration utility 10-1
 - using every statement 10-9
 - using if statement 10-6
 - using scripting language 10-5
 - using when statement 10-8
 - viewing in Configuration utility 10-2
- protection from hackers
 - using production rules 10-12
- PTR record 5-3

- Q**
- QOS coefficients A-13
- QOS equation A-13
- Quality of Service mode A-13
 - customizing 8-8
 - using dynamic ratio 8-8

- R**
- reaping A-16
- Reconfigure 3-DNS Configuration Utility command 2-6, 11-1
- redundant systems
 - displaying unit number C-18
 - fail-over C-5
 - sharing MAC addresses C-24
 - synchronizing C-4
- release notes 1-3
- remote administration 2-5
- reset command C-14
- resource records
 - A 5-2
 - CNAME 5-2
 - less common types 5-5
 - MX 5-3
 - NS 5-3
 - PTR 5-3
 - SOA 5-4
- Restart 3-DNS Configuration Utility command 2-6, 11-1
- Restart big3d command 2-5, 11-4
- Restart syncd command 2-7
- RSH C-7

- S**
- save command C-15
- saving C-15
- scripting language
 - setting up production rules 10-5
- scripts 11-1
 - 3dns_admin_start 11-1
 - 3dns_dump 11-1
 - 3dns_web_config 11-1
 - 3dns_web_passwd 11-1
 - 3dnsmaint 11-2
 - 3dprint 11-2
 - 3ndc 11-3
 - big3d_install 11-3
 - big3d_restart 11-4
 - big3d_version 11-4
 - checktrap.pl 12-5
 - edit_lock 11-4
 - edit_wideip 11-4
 - F5makekey 11-5
 - install_key 11-5
 - syncd_checkpoint 11-5
 - syncd_rollback 11-6
 - syncd_start 11-7
 - syncd_stop 11-7
- secure shell C-7
- self command C-16
- self IP address C-16
- server statement
 - 3-DNS A-18
 - about A-18
 - BIG-IP A-20
 - EDGE-FX Cache A-22
 - GLOBAL-SITE A-23
 - hosts A-23
- service monitors
 - ECV 6-1
 - ICMP C-13
 - search string in ECV 6-2
- Setting the MAC masquerade address C-10
- SNMP
 - 3-DNS OIDs 12-5
 - and probing 12-7
 - client access 12-3
 - generating traps 12-5
 - in the Configuration utility 12-7
 - MIB 12-2
 - trap configuration 12-4
- SNMP agent
 - allowing host access 12-2
 - configuration file requirements 12-2
 - configuring 12-2, 12-3
 - configuring hosts 12-10
 - denying UPD connections 12-2
 - generating traps 12-5
 - in the Configuration utility 12-6
- SNMP prober factory 12-7
- SNMP trap logs 12-5
- SOA record 5-4

- SSH C-7
 - and remote administration 2-5
 - configuring 2-5
- ssh key
 - generating 11-4
- statements
 - globals A-5
 - in wideip.conf A-3
 - server A-18
 - sync_group A-31
 - topology A-39
 - wideip A-32
- statistics
 - in 3-DNS Maintenance menu 2-2
 - using Internet Weather Map 7-3
 - viewing from command line 11-2
 - viewing with 3dpipe B-6
- Stop syncd command 2-7
- sync groups
 - archiving synchronized files 11-5
 - editing synced files 11-4
 - restoring archived files 11-6
- syncd
 - stopping or restarting 2-6
 - working with 2-6
- syncd_checkpoint script 11-5
- syncd_rollback script 11-6
- syncd_start script 2-7, 11-7
- syncd_stop 11-7
- syncd_stop script 2-7, 11-7
- synchronized files
 - and checkpoint files 11-5
 - archiving 2-5, 11-5
 - copying metrics 2-5
 - restoring from archive 11-6
 - using syncd 2-6
- syntax
 - and editing rules A-4
 - for comments A-42
 - for datacenter statement A-29
 - for globals statement A-7
 - for topology statement A-39
 - for wideip statement A-32
- syntax example
 - for datacenter statement A-30
 - for globals statement A-8
 - for server statement (3-DNS) A-19
 - for server statement (BIG-IP) A-21
 - for server statement (EDGE-FX) A-22
 - for server statement (GLOBAL-SITE) A-23
 - for server statement (host type) A-25
 - for sync_group statement A-31
 - for wideip statement A-33
- syslog utility 12-5

- T
- technical support resources 1-3
- time of day load balancing 10-10
- time synchronization 2-7
- timer settings
 - and performance 4-5
- timer values
 - about 8-17
 - and metrics collection 8-17
 - configuring 8-18
- Topology
 - using topology records 13-1
- Topology load balancing mode
 - about 13-1
 - and user-defined regions 13-5
 - configuring in pools 13-4
 - configuring in wide IPs 13-3
 - in a pool 13-1
 - in wide IPs 13-1
- topology records
 - about 13-1
 - in topology statements 13-1
 - variables 13-6
- topology statement A-39
 - variables 13-6
- triggers
 - defining 10-2
 - event-based 10-4
 - time-based 10-2
- trunk command C-17
- TTL values
 - about 8-17
 - and metrics collection 8-17
 - configuring 8-18

- U
- unit command C-18
- user administration
 - configuring 2-6
- user-defined regions 13-5
- utilities
 - bigpipe commands C-1
 - syslog 12-5

- V
- view of network 9-1
- view statistics 2-2
- virtual command C-21
- virtual server translation 4-9
- virtual servers
 - checking availability A-9
 - displaying unit number C-18
- vlan command C-21

W

- web administration C-8
- when statement
 - guidelines 10-8
 - in production rules 10-8
- wide IP production rules 10-2
- wide IPs
 - about 8-10
 - and DNS zone file management 8-10
 - configuring 2-2, 8-11
 - syntax 8-13
 - using a last resort pool 8-14
- wideip statement A-32
- wideip.conf
 - working with statements A-3
- wideip.conf file
 - about A-1
 - adding production rules 10-5
 - configuration requirements A-1
 - configuring production rules 10-5
 - ECV sub-statement A-35
 - editing 2-2, 11-4, A-1
 - syntax editing rules A-4

Z

- zone files
 - and 3-DNS Maintenance menu 2-1
 - configuring 2-2
 - managing 2-7
 - managing with NameSurfer 2-7