3-DNS® Reference Guide

version 4.6.2

MAN-0113-00

Product Version

This manual applies to version 4.6.2 of 3-DNS® Controller.

Legal Notices

Copyright

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

Trademarks

F5, F5 Networks, the F5 logo, BIG-IP, 3-DNS, iControl, GLOBAL-SITE, SEE-IT, EDGE-FX, FireGuard, Internet Control Architecture, IP Application Switch, Control Your World, PACKET VELOCITY, SYN Check, uRoam, and FirePass are registered trademarks or trademarks of F5 Networks, Inc. in the U.S. and certain other countries. All other trademarks mentioned in this document are the property of their respective owners. F5 Networks' trademarks may not be used in connection with any product or service except as permitted in writing by F5.

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 FreeBSD software: This software is provided by the FreeBSD project "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the FreeBSD project or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

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

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

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).

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

The RSA SecurID® Apache authentication module provided with this software release uses the RSA ACE/Agent® Authentication API developed by RSA Security (©2001 RSA Security Inc.).

Table of Contents

1 Introduction

2 Load Balancing

3 Topology

4 Production Rules

5
Probing and Metrics Collection

6
Administration and Monitoring

7
Configuring a Redundant System

A
3-DNS Configuration File

B
3dpipe Command Reference

C
bigpipe Command Reference

D
DNS Resource Records

E

The bigdb Database and bigdb Keys

Glossary

Index

1

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® Controller. 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 Controller. 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 following printed documentation is included with the 3-DNS unit.

- **Configuration Worksheet**
  This worksheet provides you with a place to plan the basic configuration for the 3-DNS Controller.

The following guides are available in PDF format from the CD-ROM provided with the 3-DNS Controller. These guides are also available from the home screen of the Configuration utility.

- **Platform Guide**
  This guide includes information about the physical 3-DNS unit. It also contains important environmental warnings.
- **3-DNS Administrator Guide**
  The ***3-DNS Administrator Guide*** provides examples of common wide-area load balancing solutions supported by the 3-DNS Controller. 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.

Stylistic conventions

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

Using the solution examples

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 Controller in the following locations:

- **Release notes**
  The release note for the current version of the 3-DNS Controller 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 Controller 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 Controller 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 Controller, including NameSurfer.

- **Technical support through the World Wide Web**
  The F5 Networks Technical Support web site, **http://tech.f5.com**, provides the latest technical notes, answers to frequently asked questions, updates for the Administrator Kit (in PDF format), updates for the release notes, and the Ask F5 natural language question and answer engine.

Note

All references to hardware platforms in this guide refer specifically to systems supplied by F5 Networks, Inc. If your hardware was supplied by another vendor and you have hardware-related questions, please refer to the documentation from that vendor.

2

Load Balancing

- Working with load balancing modes
- Understanding load balancing on the 3-DNS Controller
- Configuring load balancing
- Working with the ECV service monitor
- 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 Controller 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 Controller, explains the various static and dynamic load balancing modes, and then describes how to configure them.

Understanding load balancing on the 3-DNS Controller

When the 3-DNS Controller 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 Controller 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 Controller 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 Controller uses load balancing modes in two situations:

- **Load balancing among multiple pools**
  The 3-DNS Controller supports multiple pools. Configurations that contain two or more pools use a load balancing mode first to select a pool. Once the 3-DNS Controller selects a pool, the system then uses a load balancing mode to choose a virtual server within the selected pool. If the 3-DNS Controller 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 Controller 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 Controller returns the client to standard DNS for resolution.

Table 2.1 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
Drop Packet		X	X	X
Explicit IP		X	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 2.1 *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 Controller supports the following static load balancing modes:

- Drop Packet
- Explicit IP
- Global Availability
- None
- Random
- Ratio
- Return to DNS
- Round Robin
- Static Persist
- Topology

The Drop Packet, Explicit IP, 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.

Drop Packet mode

When you specify the Drop Packet load balancing mode, the 3-DNS Controller does nothing with the packet, and simply drops the request. (Note that a typical LDNS server iteratively queries other authoritative name servers when it times out on a query.) We recommend that you use the Drop Packet load balancing mode only for the fallback method. The 3-DNS Controller uses the fallback method when the preferred and alternate load balancing modes do not provide at least one virtual server to return as an answer to a query.

Explicit IP mode

When you specify the Explicit IP mode, the 3-DNS Controller returns the IP address that you specify as the fallback IP as an answer to the query. Note that the IP address that you specify is not monitored for availability before being returned as an answer. When you use the Explicit IP mode, you can specify a disaster recovery site to return when no load balancing mode returns an available virtual server. We recommend that you use the Explicit IP load balancing mode only for the fallback method. The 3-DNS Controller uses the fallback method when the preferred and alternate load balancing modes do not provide at least one virtual server to return as an answer to a query.

Global Availability mode

The Global Availability load balancing mode uses the virtual servers included in the pool in the order in which they are listed. For each connection request, this mode starts at the top of the list and sends the connection to the first available virtual server in the list. Only when the current virtual server has reached its limit settings or is 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 Controller 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 Controller uses the next available pool. If you do not have multiple pools configured, the 3-DNS Controller 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 Controller 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 2.1 shows a sample connection distribution for Ratio mode.

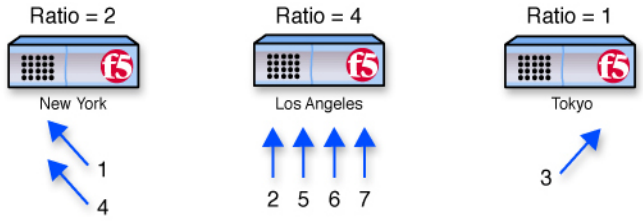

Figure 2.1 *Ratio mode*

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

Figure 2.2 *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 system or other server array manager); this is beneficial for transaction-oriented traffic, such as e-commerce shopping carts, online trading, and online banking.

Topology mode

The Topology load balancing mode allows you to direct or restrict traffic flow by adding topology records to a topology statement in the configuration file. When you use the Topology load balancing mode, you can develop proximity-based load balancing. For example, a client request in a particular geographic region can be directed to a data center or server within that same region. The 3-DNS Controller 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 Controller 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 3, *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 6, *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 Controller 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 2.3 shows a sample connection distribution pattern for the Completion Rate mode.

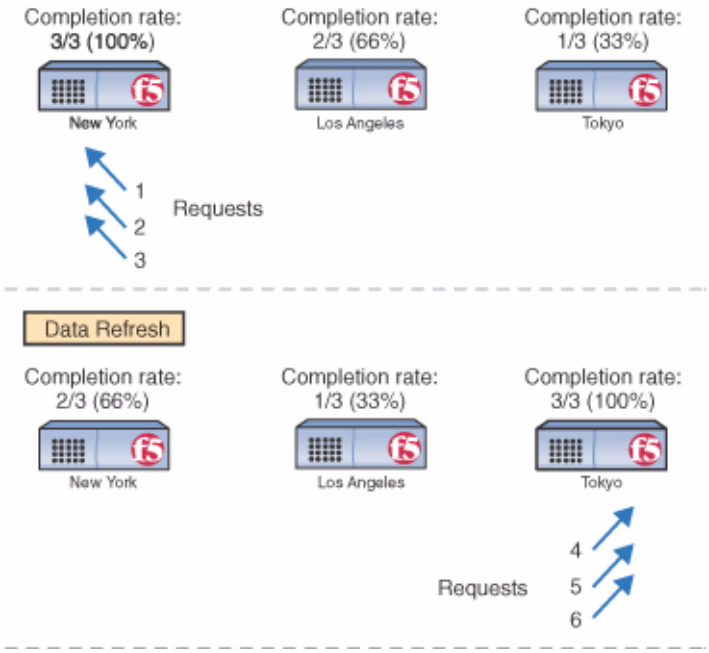

Figure 2.3 *Completion Rate load balancing mode*

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 Controller can collect the kilobytes per second metric. See *Configuring SNMP settings to probe hosts*, on page 5-19, for details on the metrics the 3-DNS Controller 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 system that currently hosts the fewest connections.

Packet Rate mode

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

Figure 2.4 shows a sample connection distribution for the Packet Rate mode.

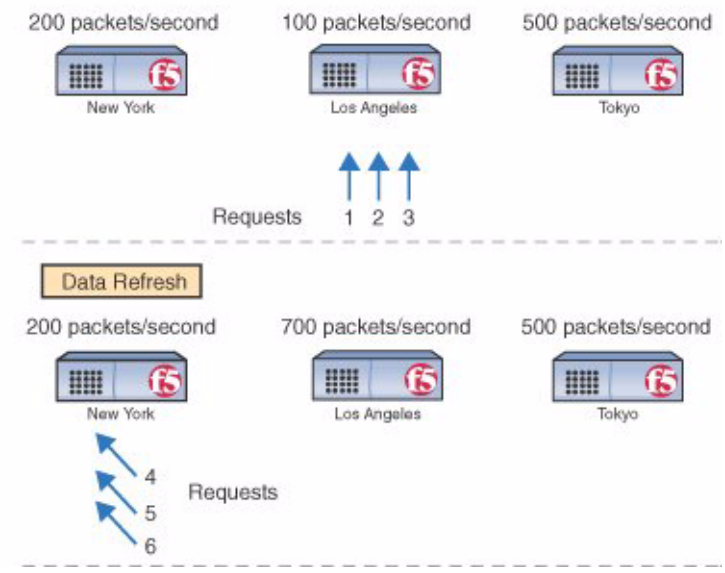

Figure 2.4 *Packet Rate mode*

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

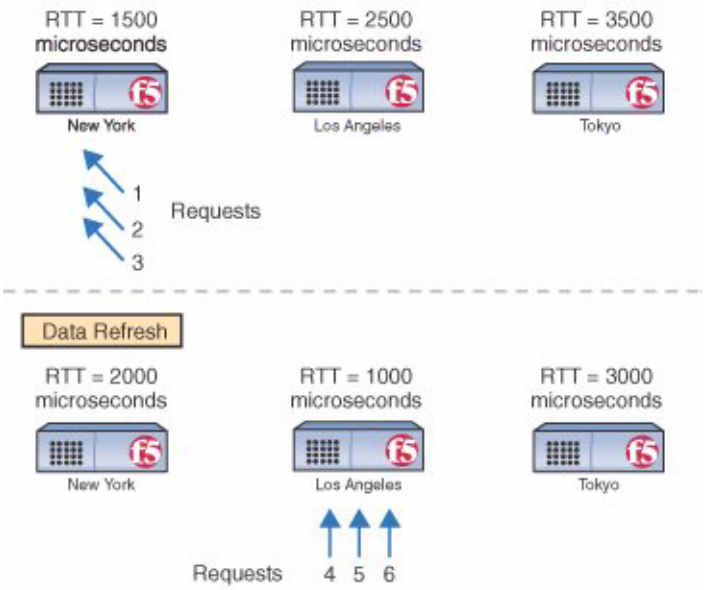

Figure 2.5 *Round Trip Times mode*

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 Controller takes into account include:

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

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

You can also configure the Quality of Service load balancing mode to use the dynamic ratio feature. With the dynamic ratio feature turned on, the Quality of Service mode becomes similar to the Ratio mode, where the

connections are distributed in proportion to ratio weights assigned to each virtual server. The ratio weights are based on the QOS scores: the better the score, the higher percentage of connections the virtual server receives.

For details about customizing the Quality of Service mode, see the ***3-DNS Administrator Guide***, Chapter 8, *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 Controller load balances using the Random mode among those virtual servers.

In the sample configuration in Figure 2.6, **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**.

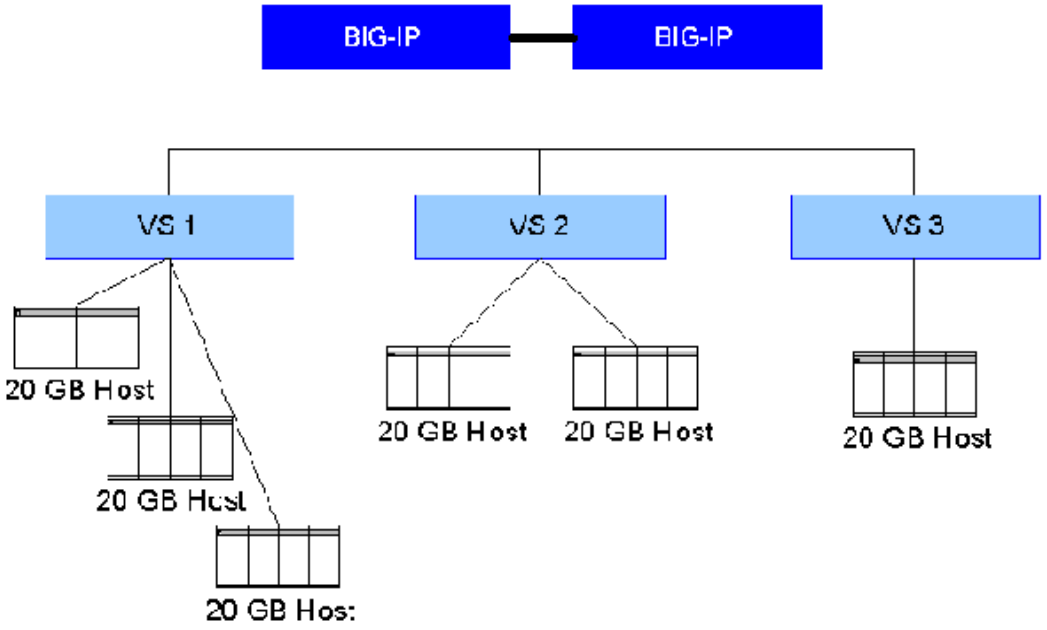

Figure 2.6 *VS Capacity load balancing mode*

Configuring load balancing

This section describes how to configure load balancing on the 3-DNS Controller. 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 Controller 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 2-20.
- **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 2-13.

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 system, 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 Controller in node mode, NameSurfer, a third-party application included with the 3-DNS Controller, 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 2-26.

Understanding pools

A wide IP contains one or more pool definitions. A ***pool*** is a group of virtual servers to which the 3-DNS Controller load balances. You can include all types of virtual servers (BIG-IP system, EDGE-FX system, 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.

◆ Note

For details on configuring a wide IP from the command line, refer to ***The wideip statement****, on page A-34.*

Using wildcard characters in wide IP names

The 3-DNS Controller 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 2.7 shows a sample **wideip** statement. This statement defines a wide IP named **mx.wip.siterequest.com**, with an alias of **mail.wip.siterequest.com**. The wide IP contains two pools, with **pool_1** receiving three times as many requests as **pool_2**. The 3-DNS Controller 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 Controller cannot resolve the request using the RTT mode, the system distributes requests using the Random load balancing mode. The 3-DNS Controller 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.siterequest.com"
   alias             "mail.wip.siterequest.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 2.7 *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 Controller 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 Controller 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 Controller uses when all other pools have reached their thresholds or are unavailable for any reason. The 3-DNS Controller uses the last resort pool only when it tries, unsuccessfully, to load balance to the virtual servers in all other configured pools.

When your network includes cache appliances that host 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 2.8 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 2.8 *Example of a last resort pool definition*

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 Controller 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 node that hosts the requested file responds appropriately to the ECV query, the 3-DNS Controller marks the virtual server where the node resides as **up**, and the controller sends the client connection request to that virtual server. If the node does not respond as expected to the ECV query, the 3-DNS Controller marks the virtual server where the node resides as **down**, and the controller does not send connections to that server.

◆ Note

*For ECV service monitors that use the HTTP protocol, the 3-DNS Controller expects the response packet from the node to contain an HTTP **200** result code. If the controller receives the **200** result code, the controller marks the virtual server as **up**. The controller marks the virtual server as **down** when it receives all other HTTP result codes.*

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. To ensure that the configuration files contain the same information as the memory cache, type the following command:

   3ndc dumpdb

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
3. Use the syntax shown in Figure 2.9 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.
4. Save and close the file.
5. Commit the changes to the configuration by typing:

   3ndc reload

```
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 2.9 *Syntax for defining ECV service monitors*

Figure 2.10 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 2.10 *Sample ECV service monitor definition*

◆ Note

*For details on the ECV options in the **wideip.conf** file, see **ECV sub-statements**, on page A-39.*

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 Controller 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 2.2.

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 2.2 *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
```

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 Controller. 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. To ensure that the configuration files contain the same information as the memory cache, type the following command:

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
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 2.11 to define global alternate and fallback load balancing methods.
5. Save and close the file.
6. Commit the changes to the configuration by typing:

   3ndc reload

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

Figure 2.11 *Configuring global alternate and fallback load balancing modes*

Figure 2.12 shows a sample **globals** statement that defines global load balancing variables.

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

Figure 2.12 *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 Controller refreshes the information.

Table 2.3 describes each TTL value, as well as its default setting.

Parameter	Description	Default
Server TTL	Specifies the number of seconds that the 3-DNS Controller uses BIG-IP system and EDGE-FX Cache metrics information for name resolution and load balancing.	60
Host TTL	Specifies the number of seconds that the 3-DNS Controller 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 Controller considers performance data for the other 3-DNS Controllers to be valid.	60

Table 2.3 *TTL values and default settings*

Parameter	Description	Default
Virtual server TTL	Specifies the number of seconds that the 3-DNS Controller uses virtual server information (data acquired about a virtual server from a BIG-IP system, EDGE-FX Cache, or host) for name resolution and load balancing.	120
Hops TTL	Specifies the number of seconds that the 3-DNS Controller considers **traceroute** data to be valid.	604800 (seven days)
Path TTL	Specifies the number of seconds that the 3-DNS Controller uses path information for name resolution and load balancing.	2400
Default TTL	Specifies the default number of seconds that the 3-DNS Controller 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 2.3 *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 Controller 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 2.4 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 Controller refreshes BIG-IP system and EDGE-FX system information.	20
Host data refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller refreshes other host machine information.	90
3-DNS data refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller retrieves performance data for other 3-DNS Controllers in the sync group.	20
Virtual server data refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller refreshes virtual server information.	30
ECV timer refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller refreshes the ECV monitor.	90
Hops data refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller 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 Controller refreshes path information (for example, round trip time or ping packet completion rate).	120

Table 2.4 *Time values and default settings*

Parameter	Description	Default
Remote nodes query	Specifies the frequency (in seconds) at which the 3-DNS Controller queries remote 3-DNS Controllers 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 Controller with a time server, leave the time tolerance at the default value of **10**. In the event that NTP fails, the 3-DNS Controller uses the **time_tolerance** variable to maintain synchronization.	10
Timer Sync State	Specifies the interval (in seconds) at which the 3-DNS Controller 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 Controller archives the paths and metrics data.	3600

Table 2.4 *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. To ensure that the configuration files contain the same information as the memory cache, type the following command:

   3ndc dumpdb

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
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 2.13, on page 2-24, to define global TTL and timer values.
5. Save and close the file.
6. Commit the changes to the configuration by typing:

   3ndc reload

```
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_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 2.13 *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 Controller 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 Controller 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 Controller 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. To ensure that the configuration files contain the same information as the memory cache, type the following command:

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
3. Locate the **wideip** statement you want to edit.
4. Add the **port_list** line as indicated in bold in Figure 2.14, on page 2-26.
5. Save and close the file.
6. Commit the changes to the configuration by typing:

   ```
   3ndc reload
   ```

```
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 2.14 *Enabling multiple ports with the* ***port_list*** *option*

An example configuration using a port list

In the example shown in Figure 2.15, 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.siterequest.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 2.15 *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 2.15, 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 Controller 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/3dns**
  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/3dns** file on your system. Using the UNIX **grep** utility, search for **3dnsd** (for example, **tail -100 /var/log/3dns | 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 Controller 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**

3

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 Controller 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 Controller)
- A data center (managed by the 3-DNS Controller)
- 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 Controller 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.

◆ Note

Note that the topology score is a value in a range that you decide, depending on the specifics of your load balancing requirements. If two records receive a score because they match the LDNS location information, the 3-DNS Controller uses the record with the higher score.

A ***topology statement*** is composed of one or more topology records. Figure 3.1 is an example of a topology statement, with two topology records, as it appears in the Configuration utility.

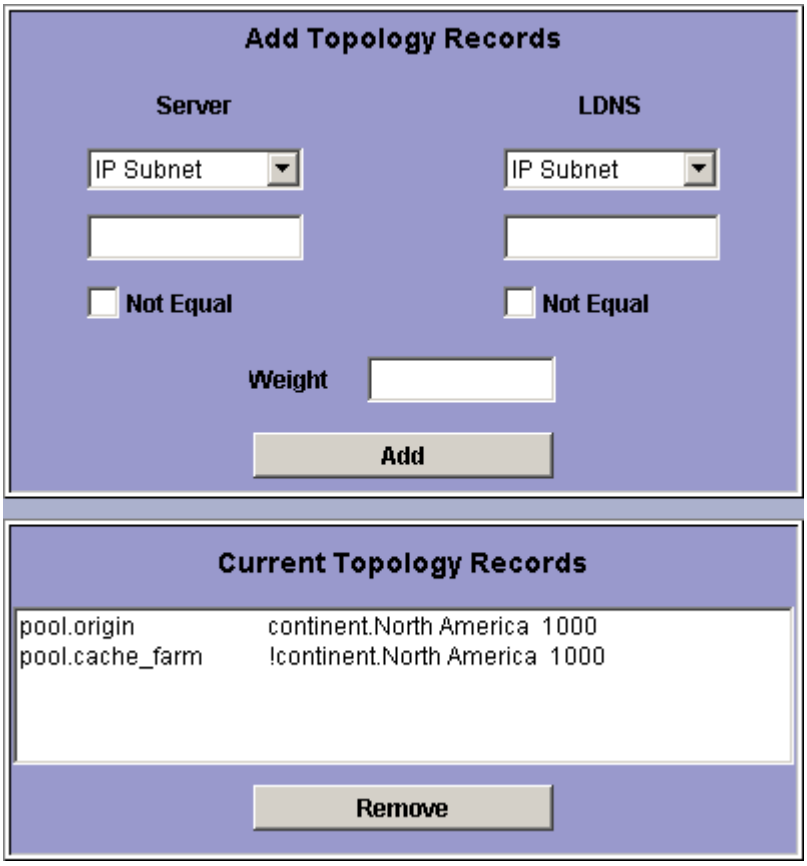

Figure 3.1 *Example of a topology statement in the Configuration utility*

Here is an explanation of how to interpret the topology statement in Figure 3.1. 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.

For example, the 3-DNS Controller receives a DNS resolution request from an LDNS located in North America. The controller evaluates the LDNS location criteria in the request against the first topology record and assigns a weight of **1000** to the record, because the location information matches. (The local DNS server is in North America.) The controller then evaluates the request against the next topology record, and assigns a score of **0** to that record because the LDNS location information does not match.

Once the controller evaluates the request against all of the topology records, it sends the request to the wide IP, pool, or subnet specified in the topology record that has the highest weight (or score). For our example, the controller 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. Type the following command to ensure that the configuration files contain the same information as the memory cache.

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
3. Locate the **wideip** statement that you want to edit.
4. Define the pool load balancing mode as **Topology**.

   ```
   pool_lbmode topology
   ```

5. Save and close the file.
6. Commit the changes to the configuration by typing:

   ```
   3ndc reload
   ```

Figure 3.2 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.siterequest.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 3.2 *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 Controller 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. Type the following command to ensure that the configuration files contain the same information as the memory cache.

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
3. Locate the **wideip** statement you want to edit.
4. Define the preferred, alternate, or fallback load balancing mode for the pool as **Topology**. The example in Figure 3.3, on page 3-8, shows a sample wide IP definition where **topology** is the preferred load balancing mode within a pool.
5. Save and close the file.
6. Commit the changes to the configuration by typing:

   ```
   3ndc reload
   ```

```
wideip {
  address 192.168.103.60
  port 80
  name "ntp.wip.siterequest.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 3.3 *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 Controller, you can create custom topology regions. By adding user-defined regions to the topology statement, the 3-DNS Controller 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 the following command to ensure that the configuration files contain the same information as the memory cache.

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
3. Add the user-defined region. You can use the syntax shown in Figure 3.4, on page 3-10, as an example.
4. Save and close the file.
5. Commit the changes to the configuration by typing:

   ```
   3ndc reload
   ```

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

Figure 3.4 *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 **/usr/local/3dns/include/net.ccdb**. 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 Controller to evaluate the best resolution option for a DNS request.

Table 3.1 *Variables used in the* ***topology*** *statement*

To add a **topology** statement to the include file **/config/3dns/include/topology.inc**, follow the format of the example in Figure 3.5.

```
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 3.5 *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 3.5.*

4

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 Controller 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 4-7, which describes the scripting language you use to configure production rules from the command line. You may also want to contact technical support 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 types of rule triggers are: a set time or time interval, a specific system event, and other conditions that trigger an action.
- **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** button, which starts the production rule wizard. The wizard prompts you to specify the various production rule options, and then 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 Remove 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 4.2, on page 4-10.
- **Wide IP production rules**
  Wide IP production rules are based on the time of day. 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 triggers for the production rule

The next step in the wizard prompts you to choose a trigger for the production rule. You can set up three basic types of triggers: time-based triggers, event-based triggers, and triggers based on other conditions. Once you review the information for the type of trigger you want to set up, go to *Choosing the action taken*, on page 4-6.

Defining triggers based on time

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.

Defining time-based triggers for global production rules

To set a time interval for a global production rule, after you have defined a name and provided a description of the production rule, complete the following steps.

To apply a set time interval variable

1. On the Select A Condition for the Production Rule page, select **Perform action according to a Time Interval**, and click **Next**.
2. On the Set the Time Interval page, in the **Seconds between Action** box, type the number of seconds you would like to elapse between each action the production rule executes, and click **Next**.

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

Defining time-based triggers for wide IP production rules

Unlike the global production rule, 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. On the Select Time Variables page, 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, and click **Next**.

   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. On the Select Time Variables page, from the Time Variable table, select **Day**.
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, and click **Next**.

   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. On the Select Time Variables page, from the Time Variable table, select **Date**.
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), and click **Next**.

   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. On the Select Time Variables page, from the Time Variable table, select **Date/Time**.
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, and click **Next**.

   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 triggers based on standard events

Standard events, such as when a name resolution process begins, can trigger only global production rules. Standard events cannot trigger wide IP production rules.

To set a standard event for a global production rule, after you have defined a name and provided a description of the production rule, complete the following steps.

To apply a standard event variable

1. On the Select A Condition for the Production Rule page, select **Perform action according to a Specific Events**, and click **Next**.
2. On the Select an Event page, in the **Select Event** box, select the event that triggers the production rule action, and click **Next**. The standard events that can trigger global production rules are described in Table 4.2, on page 4-10.

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

Defining triggers based on other conditions

Wide IP production rules contain two other conditions that you can define, in addition to time, that trigger production rules. The first condition is, you can specify a local domain name server (LDNS), and when that LDNS makes a name resolution request, the production rule triggers a certain action. The other is, you can specify the number of name resolutions that you want to occur before the production rule triggers a certain action.

To apply an LDNS variable

1. On the Select A Condition for the Production Rule page, select **Respond according to Local Domain Name Server**, and click **Next**.
2. On the Select the Local Domain Name Server page, in the **LDNS IP Address** box, type the IP address of the LDNS to which you want the business rule to apply, and click **Next**.
3. In the **LDNS Subnet Mask** box, type the subnet mask of the LDNS. Once you define the LDNS that triggers the production rule, you continue with the wizard and begin to define the production rule action.

To apply a name resolutions variable

1. On the Select A Condition for the Production Rule page, select **Respond according to Number of Name Resolutions**, and click **Next**.
2. On the Select Number of Name Resolutions page, in the **Number of Resolutions** box, type the number of name resolutions that you want to occur before the production rule triggers a certain action, and click **Next**.
   Once you define the number of name resolutions that triggers the production rule, you continue with the wizard and begin to define the production rule action.

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 Controller 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 Controller).

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 **3dscript.** Production rules must have the following attributes in **3dscript**:

- 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 **3dscript** language manages and executes production rules according the following guidelines:

- The **3dscript** 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 **3dscript** language supports event-driven execution of production rules using the **when** statement. You can use the **when** statement only in global production rules.
- The **3dscript** 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 **3dscript** language.

Working with the if statement

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

The **if** statement is a standard statement that defines a 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 Controller.
- 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 (II) 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 4.1 *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 global production rule event triggers are described in Table 4.2.

Event triggers	Description
ResolveNameBegin	The production rule takes action each time the 3-DNS Controller receives a new resolution request.
ResolveNameEnd	The production rule takes action each time the 3-DNS Controller 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 Controller receives a SIGINT command.
SIGHUP	The production rule takes action each time the 3-DNS Controller receives a SIGHUP command.
ReapPaths	The production rule takes action each time the 3-DNS Controller reaps obsolete path information.
CRC_Failure	The production rule takes action each time iQuery communication on the 3-DNS Controller experiences a CRC failure.
DownServer	The production rule takes action each time the 3-DNS Controller detects that another 3-DNS, BIG-IP, or host server becomes unavailable.
DownVS	The production rule takes action each time the 3-DNS Controller 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 4.2 *Global 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. You can use an **every** statement only for a global production rule, and it must adhere to the following guidelines:

- Specify the **every** statement at the top scope of the **wideip.conf** file, after the wide IP definition(s) and before the **topology** statement.
- Specify multiple **every** statements in the same scope.
- You cannot nest **every** statements.

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 4.3 *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 4.3 *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 Controller 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 4.1 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.siterequest.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 4.1 *Production rule example for 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 4.2 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.siterequest.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 4.2 *Production rule example for 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 4.3 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 4.3 *Production rule example for sending a hacker to a specific server*

A related example, shown in Figure 4.4, illustrates a production rule that deals with attacks against iQuery communications. The production rule warns you if the 3-DNS Controller detects a hack attempt against the iQuery protocol, based on a communication failure.

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

Figure 4.4 *Production rule example for detecting an iQuery failure due to potential attack*

5

Probing and Metrics Collection

- Overview of probing and metrics collection
- Working with the big3d agent
- Working with SNMP on the 3-DNS Controller
- Working with access control lists

Overview of probing and metrics collection

The 3-DNS Controller can collect server availability and capacity information from any server or appliance in the network using the **big3d** agent. The **big3d** agent can also collect path information (router hops, round trip times, and completion rates) from local DNS servers. This data collecting is known as ***probing***. The 3-DNS Controller uses the **big3d** agent to probe all BIG-IP systems and EDGE-FX systems in the controller's network. The **big3d** agent uses SNMP to probe any hosts or routers in the controller's network. Alternately, if you do not want the **big3d** agent to probe certain servers, for example, America Online (AOL) local DNS servers, you can create access control lists (ACLs).

Working with the big3d agent

The **big3d** agent collects performance information on behalf of the 3-DNS Controller. The **big3d** agent runs on 3-DNS Controllers, BIG-IP systems, and EDGE-FX 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. If you turn off the **big3d** agent on a server, the 3-DNS Controller can no longer check the availability of the server or its virtual servers, and the statistics screens display the status as **unknown** (blue ball).

◆ Note

*We recommend that you have a **big3d** agent running on at least one system in each data center in your network.*

Collecting path data and server performance metrics

A **big3d** agent collects the following types of performance information used for load balancing. The **big3d** agent broadcasts this information to all 3-DNS Controllers in your network.

- **Network path round trip time**
  The **big3d** agent calculates the round trip time for the network path between the agent's data center and the client's LDNS server that is making the resolution request. The 3-DNS Controller uses round trip time to determine the best virtual server to answer the request when a pool uses a dynamic load balancing mode, such as Round Trip Time, or Quality of Service.

- **Network path packet loss**
  The **big3d** agent calculates the packet completion percentage for the network path between the agent's data center and the client's LDNS server that is making the resolution request. The 3-DNS Controller uses the packet completion rate to determine the best virtual server to answer the request when a wide IP or pool uses either 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 system transitions (router hops) between the agent's data center and the client's LDNS server. The 3-DNS Controller uses hops to determine the best virtual server to answer the request when a pool uses the Hops or the Quality of Service load balancing modes.
- **Server performance**
  The **big3d** agent returns server metrics, such as the packet rate, for BIG-IP systems or SNMP-enabled hosts. The 3-DNS Controller uses packet rate to determine the best virtual server to answer the request when a pool uses the Packet Rate, KBPS, Least Connections, or Quality of Service load balancing modes.
- **Virtual server availability and performance**
  The **big3d** agent queries virtual servers to verify whether they are **up** and available to receive connections, and uses only those virtual servers that are **up** for load balancing. The **big3d** agent also determines the number of current connections to virtual servers that are defined on BIG-IP systems or SNMP-enabled hosts. The 3-DNS Controller uses the number of current connections to determine the best virtual server when a pool uses the Least Connections or VS Capacity load balancing mode.

Setting up data collection with the big3d agent

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

- **Installing big3d agents on BIG-IP systems and EDGE-FX systems**
  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 system and EDGE-FX system in the network. See the release notes provided with the 3-DNS software for information about which versions of the BIG-IP software and the EDGE-FX software the current **big3d** agent supports. For details on installing the **big3d** agent, see *Installing the big3d agent*, on page 5-3.
- **Specifying which factories a specific big3d agent manages**
  When you define 3-DNS Controllers, BIG-IP systems, and EDGE-FX 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 5-3.

- **Setting up communications between big3d agents and other systems**
  Before the **big3d** agents can communicate with the 3-DNS Controllers 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 Controllers in the network. These planning issues are discussed in *Setting up communication between 3-DNS Controllers and other servers*, on page 5-6.

Installing the big3d agent

You can use the 3-DNS Maintenance menu to easily install the **big3d** agent on the BIG-IP systems and EDGE-FX systems in your network.

To install the big3d agent from the command line

1. Log on to the 3-DNS Controller 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 the DNS_DOT, DNS_REV, ICMP, TCP, or UDP protocols. (Note that the prober factory uses the protocols in the listed order.) The prober 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 calculates the round trip time and packet loss rate 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 *Working with the ECV service monitor*, on page 2-17.

The standard configuration specifies that each 3-DNS system, BIG-IP system, and EDGE-FX system 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. Note also that the controller dynamically increases the number of factories if needed.

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

1. In the navigation pane, click **Servers**, and then click **EDGE-FX**.
   The EDGE-FX List screen opens.
2. In the list, click the name of the EDGE-FX system that you want to modify.
   The Modify EDGE-FX 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.

Understanding the data collection and broadcasting sequence

The **big3d** agents collect and broadcast information on demand. The principal 3-DNS Controller 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 Controllers running in the network, including the principal 3-DNS system that issued the request.

Tracking LDNS probe states

The 3-DNS Controller tracks the state of path data collection for each LDNS that has ever requested a name resolution from the system. Table 5.1 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 5.1 *Probe states for individual client LDNS servers*

Evaluating big3d agent configuration trade-offs

You must run a **big3d** agent on each BIG-IP system, 3-DNS system, and EDGE-FX system in your network if you use dynamic load balancing modes (those that rely on path data) on the 3-DNS Controller. (For information about dynamic load balancing, see *Using dynamic load balancing modes*, on page 2-7.) 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 Controller.

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 Controllers and other servers

In order to copy **big3d** agents from a 3-DNS Controller to BIG-IP systems and EDGE-FX systems, the 3-DNS Controller must be able to communicate with the other 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 **setup**, and press Enter.
   The Setup utility opens.
2. From the menu, choose **(U) Configure RSH**, and press Enter.
3. Follow the onscreen instructions to enable the **rsh** and **rcp** tools.

Important

*As of 3-DNS Controller, version 4.5, we no longer distribute non-crypto systems. You may, however, want to enable **rsh** if you have older, non-crypto systems in your network.*

Table 5.2 shows the ports and protocols that a 3-DNS system uses to communicate with crypto and non-crypto BIG-IP systems and EDGE-FX systems.

From	To	Protocol	From Port	To Port	Purpose
Crypto 3-DNS Controllers	Crypto BIG-IP systems, crypto EDGE-FX systems	TCP	<1024	22	SSH/SCP
Non-crypto 3-DNS Controllers	Non-crypto BIG-IP systems, Non-crypto EDGE-FX systems	TCP	<1024	514	RSH/RCP
Crypto 3-DNS Controllers	Non-crypto BIG-IP systems, Non-crypto EDGE-FX systems	TCP	<1024	514	RSH/RCP
Non-crypto BIG-IP systems, Non-crypto EDGE-FX systems	Crypto 3-DNS Controllers	N/A	N/A	N/A	N/A

Table 5.2 *Communications between 3-DNS Controllers, BIG-IP systems, and EDGE-FX systems*

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 other crypto systems, however, crypto 3-DNS Controllers use Blowfish encryption after the iQuery encryption key has been copied to all crypto 3-DNS Controllers, BIG-IP systems, and EDGE-FX 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 throughout the network and 3-DNS Controllers. The ports used by iQuery traffic change, depending on whether the traffic is inbound from the **big3d** agent or outbound from the 3-DNS Controller.

Table 5.3 shows the protocols, ports, and iQuery settings for both inbound and outbound iQuery communications between 3-DNS Controllers and **big3d** agents distributed in your network.

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

Table 5.3 *Communication protocols and ports between 3-DNS Controllers 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 to 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 5.4 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 5.4 *Communication protocols and ports between* ***big3d*** *agents and SNMP agents on hosts*

If you run a **big3d** agent on a 3-DNS system or a BIG-IP system, 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 network address translation when there is a firewall in the path between the **big3d** agent and the 3-DNS Controller. The firewall translates only the packet headers, not the payloads.

The virtual server translation option resolves this issue. When you configure address translation for virtual servers, 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 Controller reads the IP address out of the packet payload, rather than out of the packet header.

In the example configuration shown in Figure 5.1, a firewall separates the path between a BIG-IP system running a **big3d** agent, and the 3-DNS Controller. The packet addresses are translated at the firewall. However, addresses within the iQuery payload are not translated, and they arrive at the BIG-IP system in their original states.

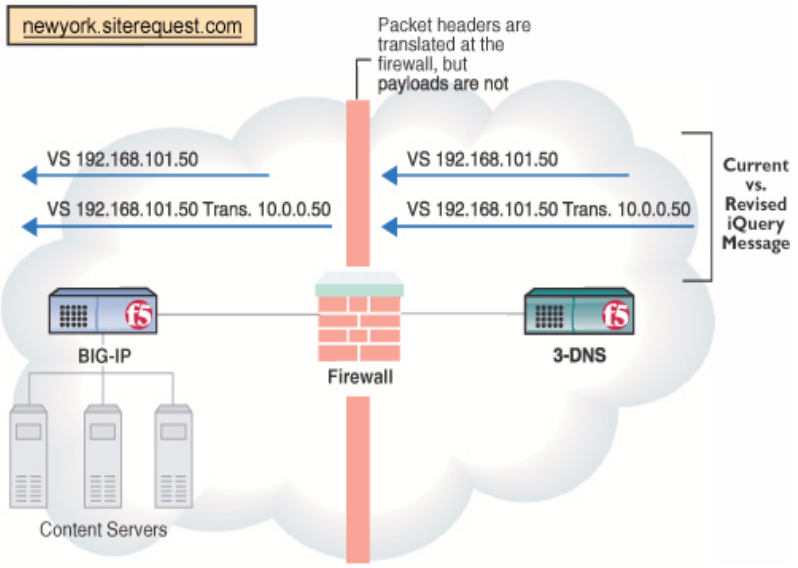

Figure 5.1 *Translating packet address through the firewall*

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

Table 5.5 shows the ports on which the 3-DNS Controller 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 5.5 *DNS communications on the 3-DNS Controller*

Table 5.6 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 5.6 *Communications between* ***big3d*** *agents and local DNS servers*

Working with SNMP on the 3-DNS Controller

The 3-DNS Controller ships with a customized simple network management protocol (SNMP) agent and management information base (MIB). This section 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 Controller 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.

Important

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

Configuring SNMP on the 3-DNS Controller

To use SNMP on the 3-DNS Controller, 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.pl** script

Note

*If you are configuring the 3-DNS Controller module on a BIG-IP system, you configure any SNMP settings using the BIG-IP Configuration utility. For information about working with SNMP on a BIG-IP system, refer to the **BIG-IP Reference Guide**.*

Downloading the MIBs

The 3-DNS Controller includes a proprietary 3-DNS SNMP MIB. This MIB is specifically designed for use with the 3-DNS Controller. 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 **/usr/local/share/snmp/mibs** directory on the controller, 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

Before using the SNMP agent, you need to make changes to several configuration files on the 3-DNS Controller. You can make these changes either by using the Configuration utility or by modifying the files from the command line. Once you change these configuration files, you must restart the SNMP agent. The files are discussed in the following sections.

/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.

WARNING

*The **/etc/hosts.allow** file must contain the following entry, which is in the file by default: **snmpd : 127.0.0.1**. If you remove this entry, the 3-DNS Controller cannot properly poll using SNMP.*

Adding a list of specific IP addresses to the etc/hosts.allow file

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
```

Adding an address range to the etc/hosts.allow file

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 syntax 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 5-18.*

/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/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.

The following list contains a few of the necessary variables:

- **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* ***trapsink*** *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 5-18.*

/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 5.2.

```
# 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 5.2 *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 Controller include:

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

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

/etc/syslog.conf

To generate traps, you must configure **syslog** to send syslog lines to **checktrap.pl**. If the syslog lines match a 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 Controller. The default is **/usr/local/bin/snmptrap**.

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

- **Date remova**l
  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 specific network address or an address range from which SNMP requests are accepted. The Configuration utility adds the client access entries to the **etc/hosts.allow** file.
- **System information**
  You can define a system contact, a machine location, and a community string. The Configuration utility adds the system information to the **/etc/snmpd.conf** file.
- **Trap configuration**
  You can enter a trap sink and a trap community. The Configuration utility adds the trap configuration information to the **/etc/snmpd.conf** file.

Note

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

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.
3. For help on configuring the SNMP settings, click **Help** on the toolbar.

WARNING

*The **/etc/hosts.allow** file must contain the following entry, which is in the file by default: **snmpd : 127.0.0.1**. If you remove this entry, the 3-DNS Controller cannot properly poll using SNMP. When you use the Configuration utility to configure the systems's SNMP properties, this address is already listed in the **Allow List** box.*

Configuring SNMP settings to probe hosts

After defining a host server or router, you need to configure its SNMP settings if you want to use SNMP to probe that host or router. Remember that you must first set up at least one SNMP prober factory on any 3-DNS Controller, BIG-IP system, or EDGE-FX system that runs the **big3d** agent and is in the same data center as the host or router.

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

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

The 3-DNS Controller gathers metrics for BIG-IP systems, EDGE-FX Caches, and several host servers. Refer to Table 5.7 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 system	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	
NetApp® appliance	X	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 5.7 *Server types and the metrics collected by the 3-DNS Controller*

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. Type the following command to ensure that the configuration files contain the same information as the memory cache.

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
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.
7. Save and close the file.
8. Commit the changes to the configuration by typing:

   ```
   3ndc reload
   ```

Figure 5.3 shows the SNMP syntax for a host server, in bold.

```
server {
  type host
  box <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 | win2kserv | ciscold | ciscold2 | ciscold3
| foundry | arrowpoint | alteon | cacheflow | netapp>
    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 5.3 *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.

Working with access control lists

With access control lists (ACLs), you can block probing for members of the ACL when you use dynamic LDNS probing on the 3-DNS Controller. Table 5.8 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 5.8 *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 **/config/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 5.4 is an example of the syntax for a **region.ACL** file with definitions for the two ACL types.

```
}
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 5.4 *Sample* ***region.ACL*** *file*

6

Administration and Monitoring

- Monitoring and administration utilities provided on the 3-DNS Controller
- Managing user accounts
- Managing the SSH Console
- Overview of the Network Map
- Viewing system statistics
- Overview of the Internet Weather Map
- Working with command line utilities
- Configuring Email
- Using a serial terminal with the 3-DNS Controller
- Shutting down the 3-DNS Controller

Monitoring and administration utilities provided on the 3-DNS Controller

The 3-DNS Controller provides several utilities for monitoring and administration. You can perform configuration tasks and monitor system statistics for all components of the 3-DNS Controller with these utilities:

- **Configuration utility**
  The Configuration utility is a browser-based application you can use to configure and monitor the 3-DNS Controller. The Configuration utility supports Netscape® Navigator, version 4.7x, and Microsoft® Internet Explorer, version 5.0, 5.5, or 6.0.
- **Setup utility**
  The Setup utility is a menu-driven command line utility that you can use to configure many of the platform settings for the 3-DNS Controller. You can also use the browser-based version of the Setup utility for the initial configuration of the 3-DNS Controller. If you are using the Setup utility to make changes to your existing configuration, we recommend that you use the command line version of the utility. To access the Setup utility from the command line, type **setup**.
- **3-DNS Maintenance menu**
  The 3-DNS Maintenance menu is a command line utility you can use to configure the 3-DNS Controller. Use the 3-DNS Maintenance menu to simplify certain tasks such as updating the **big3d** agent and configuring **ssh** access. To access the 3-DNS Maintenance menu from the command line, type **3dnsmaint**.
- **MindTerm SSH Client**
  The MindTerm SSH Client is a secure shell tool with which you can use, from the Configuration utility, the command line utilities from a web browser.
- **Network Map**
  The Network Map is an interactive screen, in the Configuration utility, where you can view your physical and logical configurations simultaneously.
- **Statistics screens**
  Using the Statistics screens in the Configuration utility, you can view a myriad of performance and metrics details about the 3-DNS Controller, the servers and the virtual servers it manages, and the load balancing it performs.
- **3dpipe utility**
  Using the **3dpipe** utility, you can perform the following tasks, at the command line:
  - View lists of configured data centers, servers, virtual servers, wide IPs, and pools
  - View the status (enabled or disabled) of configured data centers, servers, virtual servers, wide IPs, and pools

- Enable configured data centers, servers, virtual servers, wide IPs, and pools
- Disable, for a specific time period, configured data centers, servers, virtual servers, wide IPs, and pools
- View summary statistics for the 3-DNS Controller itself

◆ **bigpipe** utility
You can use the **bigpipe** utility to maintain and monitor the platform components of the 3-DNS Controller, including VLANs, interfaces, and self IP addresses. Review Appendix C, *bigpipe Command Reference*, for a complete explanation of working with the **bigpipe** utility.

Managing user accounts

When you run the Setup utility for the first time to configure your base network, the 3-DNS Controller automatically creates two special user accounts--**root** and **admin**. As an option, you can also specify within the Setup utility that you want the 3-DNS Controller to create a third account, **support**, which gives F5 Networks support personnel access to your system. For information on using the Setup utility to create the **root**, **admin**, and **support** accounts, see the ***3-DNS Administrator Guide,*** Chapter 3, *Using the Setup Utility.*

Once the Setup utility has created these accounts, you will most likely want to create additional administrative accounts and assign various system access levels, or user roles, to them, on an ongoing basis.

The remainder of this section addresses the following topics:

- Understanding user roles
- Creating and authorizing local user accounts
- Creating and authorizing remote user accounts
- Managing passwords for local accounts
- Managing system accounts

◆ Note

If you are running the 3-DNS module on a BIG-IP system, you manage user accounts using the BIG-IP Configuration utility. See the ***BIG-IP Reference Guide*** *for more information.*

Understanding user roles

Users who have user roles assigned to them fall into one of two categories: fully-privileged users, or restricted users. The following sections describe these user-role categories.

Fully-privileged users

Fully-privileged users are those who have full access to a 3-DNS Controller for administration purposes. When creating accounts for users to whom you want to grant full privileges, you can choose one of three different roles. The role you choose for a user depends on the type of user interface that the user will use to administer the 3-DNS Controller. Because each role has full access to the 3-DNS Controller, users with these user roles have privileges to change their own roles or other users' roles.

The roles for fully-privileged users are:

- **Full Web Read/Write**
  This access level provides the user with full access to all administrative tasks. Users with this access level can access the 3-DNS Controller through the Configuration utility and iControl, but not through the command line interface.
- **CLI + Full Web Read/Write**
  This access level provides the user with full access to all administrative tasks. Users with this access level can access the 3-DNS Controller through all external interfaces--the Configuration utility, the command line interface, and the iControl interface.
- **CLI**
  This access level provides the user with full access to all administrative tasks, using the command line interface.

Important

The three roles listed above all grant the same level of user access, that is, full access to the 3-DNS Controller. Thus, these roles are not intended as a way to restrict administrative access; rather, they are provided strictly as a way to define the method of user access, for administrative convenience.

Restricted users

Restricted users are those whose administrative access to a 3-DNS Controller is limited. When creating accounts for users to whom you want to restrict access, you can choose one of three different roles, where each role represents a different level of access to the 3-DNS Controller. The role you choose depends on the level of restricted access that you want to grant to the user. The roles for restricted users are:

- **Partial Web Read/Write**
  This access level allows the user to view information and to change the status of objects in the configuration to either **enabled** or **disabled**. Users with this access level can access the 3-DNS Controller through the Configuration utility only.
- **Web Read Only**
  This access level allows the user to view information using the Configuration utility only. Users with this access level do not have access to **Add** buttons, certain toolbar or tab items, **Apply** buttons, **Update** buttons, or **Remove** buttons.

- **None**
  This access level is the default access level, and prevents the user from accessing the 3-DNS Controller altogether.

The procedure that you use to create and manage user accounts depends on whether you have configured user authentication to use either the local LDAP database that resides on the 3-DNS Controller, or an external (remote) server. The following sections describe how to assign access levels based on these two different authentication scenarios.

Note

*The **root**, **admin**, and **support** accounts require special consideration when managing them. For information on managing these accounts, see* ***Managing system accounts**, on page 6-8.*

Creating and authorizing local user accounts

When you are using the local LDAP database on the 3-DNS Controller to authenticate users, your 3-DNS administrative accounts (including user names and passwords) are created and stored in the local LDAP database on the 3-DNS Controller, using the Configuration utility. Then you use the Configuration utility to assign a level of access, or user role, to each user account. Upon user authentication, the 3-DNS Controller checks the local LDAP database to determine the access level for that user. An exception to this is the **root** account, which is stored in the UNIX **/etc/passwd** file, rather than in the local LDAP database.

You assign access levels to users at the time that you create their user accounts or by changing the properties of an existing account.

Creating, changing, and deleting user accounts

You can use the Configuration utility to create new user accounts on the 3-DNS Controller. For each user account that you create, you can assign one level of access control.

To display a list of existing user accounts using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This displays a list of all existing local user accounts.

Note

*The Configuration utility only displays those accounts that are stored in the local LDAP database. Thus, the **root** account does not appear in the list of user accounts, given that the account is stored elsewhere.*

To create a user account using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This displays a list of all local user accounts, except for the **root** account.
3. Click the **Add** button.
4. In the **Add User** section, type the following information:
   - **User ID**
     Type the user ID you want to assign the user.
   - **Password**
     Type the password you want to assign the user.
   - **Retype Password**
     Retype the password you want to assign the user.
5. Select an access level for the user.
6. Click **Done**.

To change the properties of a user account using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This displays a list of all local user accounts, except for the **root** account.
3. Click a user account name.
   This displays the properties of that account.
4. Change the password, or select a new access level for the account.
5. Click **Apply**.

WARNING

*If you have a redundant system configuration and you change the password on the **admin** account, you must also change the password on the second unit in the redundant system, to ensure that the **bigpipe config sync** command operates correctly.*

To delete a user account using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This lists the user roles currently assigned to local user accounts.
3. In the **Local Users** box, locate a user name for which you want to delete a user role, and click the Remove button.
   Note that you cannot delete the **admin** user account.

Creating and authorizing remote user accounts

When you are using a remote LDAP, RADIUS or RSA SecurID authentication server, you create and store your 3-DNS administrative accounts (including user names and passwords) on that remote server, using the mechanism supplied by that server's vendor.

To configure user authorization in this case, you use the Configuration utility to assign a specific access level, or user role, to each remote user account. This access information is then stored in the 3-DNS Controller's local LDAP database. When a user, whose account information is stored remotely, logs into the 3-DNS Controller and is granted authentication, the 3-DNS Controller then checks its local LDAP database to determine the access level that is assigned to that user.

If no user role is assigned to a remote user account, then the 3-DNS Controller assigns access based on a role called the Default Role. Using the Configuration utility, you can set the access level for the Default Role.

The following sections describe the procedures for assigning user roles to remote user accounts.

To display a list of user roles for remote accounts using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This displays the **Remote User Roles** box, which lists the remote user accounts to which you have assigned an access level, as well as the Default Role and its access level. Also displayed is the **Local Users** box, showing the **admin** account, which is always stored locally on the 3-DNS Controller.

Important

Any user account that has not been assigned a remote user role automatically inherits the access level assigned to the Default Role.

To assign a user role for a remote account using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
3. Click the **Add User Role** button.
   The Add User screen opens.
4. In the **User ID** box, type a user name that is stored on your remote authentication server.
5. In the **Access Level** box, select an access level to assign to that user.
6. Click **Done**.

To change a user role for a remote account using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This lists the user roles currently assigned to remote user accounts.
3. In the **Remote User Roles** box, click a user name.
   This displays the user role properties for that user account.
4. In the **Access Level** box, select a different access level.
5. Click **Apply**.

To delete a user role for a remote account using the Configuration utility

1. In the navigation pane, click **System Admin**.
2. Click the User Administration tab.
   This lists the user roles currently assigned to remote user accounts.
3. In the **Remote User Roles** box, locate a user name for which you want to delete a user role and click the Delete button.

Managing passwords for local user accounts

Sometimes, the users who have accounts stored in the local LDAP database might need to change their passwords. Users can change their passwords by accessing the User Administration screen of the Configuration utility, and then displaying the properties of their user accounts.

This method of changing a password applies not only to the user accounts you create from within the Configuration utility, but also to the **admin** and **support** accounts that the Setup utility created when you configured your base network.

For the procedure on changing passwords for locally-stored user accounts, see *To change the properties of a user account using the Configuration utility*, on page 6-5.

◆ Note

*To change the password for the **root** account, you must re-run the Setup utility. For more information, see the following section.*

Managing system accounts

As previously described, the Setup utility automatically creates three system accounts--**root**, **admin**, and **support**. Only the support account is optional.

These accounts must be managed in the following ways:

- **The root account**
  The **root** account is defined in the **/etc/passwd** file on the 3-DNS Controller, and therefore does not reside in either the local LDAP database or a remote LDAP database. To initially create the **root** account and set its password, you run the Setup utility. To change the **root** account password later, you must re-run the Setup utility. Because the **root** account does not reside in the local or a remote LDAP database, it does not appear on the User Administration screens of the Configuration utility. The access level for this account is fixed during creation and cannot be changed.
- **The admin account**
  The **admin** account is defined in the local LDAP database on the 3-DNS Controller. To initially create the **admin** account and set its password, you run the Setup utility. To change its password later, you use the Configuration utility's User Administration screens. Note, however, that due to redundant system considerations, you must change the password on both units of the redundant system configuration, and you cannot delete the password for this account. The access level for this account is fixed during creation and cannot be changed, except when performing an upgrade.
- **The support account**
  The **support** account is defined in the local LDAP database on the 3-DNS Controller. To initially create the **support** account and set its password, you run the Setup utility. Unlike the **root** and **admin** accounts, however, creating the **support** account is optional. To change the password and access level for this account later, you use the Configuration utility's User Administration screens.

Managing the SSH Console

An SSH console gives you the ability to use a command line interface to securely manage your local 3-DNS Controller. You can either use the MindTerm SSH console that is available in the navigation pane of the Configuration utility, or you can download a different SSH console from the home screen of the Configuration utility.

Using the MindTerm SSH Client

With the MindTerm SSH Client, you can open an SSH session to the 3-DNS Controller from the Configuration utility. The 3-DNS Controller uses the MindTerm SSH Client to enable secure command line administration from a web browser. You can perform any of the command line tasks in a popup console screen.

WARNING

The MindTerm SSH client requires a Java virtual machine to operate. If you are unable to run the MindTerm SSH client, make sure that you have a Java virtual machine installed, and that your browser has Java enabled in the Preferences, or Options, section. For more information on Java virtual machines and download options, visit your web browser manufacturer's web site.

To open the MindTerm SSH Client using the Configuration utility

1. In the navigation pane, click **MindTerm SSH Client**.
   A popup screen opens.
2. When you see the command prompt, press Enter.
3. Log in to the 3-DNS Controller as you normally would.

Note

When you use the MindTerm SSH Client, you can administer only the local 3-DNS Controller. If you wish to administer remote systems, you do so using an SSH or Telnet session from the command line on the local 3-DNS Controller. For information about installing an SSH client on the administrative workstation, see the following section.

Downloading an SSH client to your administrative workstation

From 3-DNS units that support encrypted communications, you can download the SSH client to your administrative workstation in preparation for remote command line access. In addition to running 3-DNS Controller command line utilities, you can also use the SSH suite for file transfer to and from the 3-DNS Controller, as well as for remote backups.

The SSH client is available for both Windows® and UNIX® platforms, and you can download your preferred client either from the web server or using an FTP connection. You can find detailed information about the SSH client in the documentation provided on the web server, or on the Documentation and Software CD-ROM.

Downloading the SSH client from the web server

1. Connect to the 3-DNS Controller using **https://** rather than **http://** in the URL.
2. In the Additional Software Downloads section, click the **SSH Clients** link.
3. From the SSH Clients page, you can select the SSH Client appropriate to your operating system.

Note

You can also download the SSH clients from the Software and Documentation CD, or from the Ask F5 web site, ***http://tech.f5.com****.*

Setting up an SSH client on a Windows 95 or Windows NT workstation

The SSH client installation file for Windows platforms is compressed in ZIP format. You can use standard ZIP tools, such as PKZip or WinZip to extract the file.

To unzip and install the SSH client

1. Log on to the Windows workstation.
2. Navigate to the directory to which you transferred the installation file. Run **PKZip** or **WinZip** to extract the files.
3. The set of files extracted includes a Setup program. Run the Setup program to install the client.
4. Start the SSH client.
5. In the SSH Client window, from the Edit menu choose **Properties**. The Properties dialog box opens.
6. In the Connection tab, in the Remote Host section, type the following items:
   - In the **Host Name** box, type the 3-DNS Controller IP address or host name.
   - In the **User Name** box, type the root user name.
7. In the Options section, check **Compression** and set the Cipher option to **Blowfish**.
8. Click the **OK** button.

Overview of 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 configuration
- Use the navigational tools to modify your network configuration
- View the enabled/disabled state of the various objects in your network

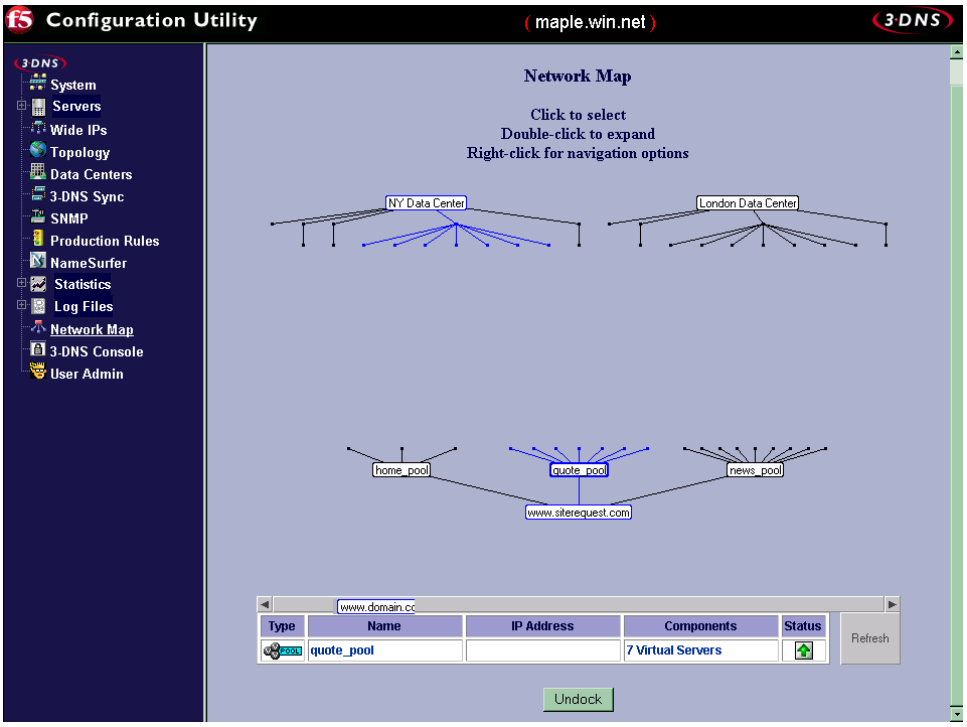

Figure 6.1 *Example screen of the Network Map in the Configuration utility*

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 6.1, the wide IP pool, **quote_pool**, is made up of virtual servers on a BIG-IP system 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.

◆ Note

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, also highlighted are the wide IPs that contain a virtual server which belongs to the servers in the selected data center. 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.

Managing your configuration with the Network Map

The Network Map is a dynamic, illustrative map of the physical and logical components of your network. The Network Map lets you see how the data centers, servers, and virtual servers you configured are mapped to the wide IPs and pools you configured for load balancing. You can also make changes to your configuration from the Network Map, using the following options:

- You can double-click any object name on the Network Map to expand the object.
- You can right-click any object name to view a popup menu of configuration options for that object.

To manage your configuration using the Network Map

1. In the navigation pane, click **Network Map**.
   The Network Map screen opens.
2. To see the relationships between the components, double-click the component. The tree expands and the component is highlighted (in blue).
3. To modify a component, right-click the component to view a popup menu, then select the item you want to change.
4. You can also click the name of the component in the status bar in the lower portion of the screen to edit the component's configuration.

For more information on the features of the Network Map, click **Help** on the toolbar.

WARNING

The Network Map requires a Java virtual machine to operate. If you are unable to view the Network Map, make sure that you have a Java virtual machine installed and that your browser has Java enabled in the Preferences, or Options, section. For more information on Java virtual machines and download options, visit your web browser manufacturer's web site.

Viewing system statistics

Using the Configuration utility, you can view current statistics about the following objects in the configuration:

Statistics screen	Description
Summary	Provides information about the 3-DNS Controller itself.
Globals	Provides information on the global settings for the 3-DNS Controller.
System Graphs	Provides CPU usage and memory usage statistics for the 3-DNS Controller in a graphical format so that you can view changes and trends in statistics over time.
Metrics	Provides performance information for the servers, virtual servers, and pools you have configured.
Links	Provides information about the router links in the network.
P95 Billing	Provides information about the average actual link utilization compared to purchased bandwidth.
Disabled	Provides information on the servers, virtual servers, wide IPs, pools, and data centers that are currently disabled.
Requests	Provides information on the virtual connections between local DNS servers and virtual servers for given wide IPs in the network.
Data Centers	Provides information on the data centers in your network.
Sync Group	Provides information on the 3-DNS Controllers that are in the same sync group as the controller that you are looking at.
Wide IPs	Provides information on the wide IPs, pools, and virtual servers in the pools.
ECV	Provides performance information for any ECV health monitors you have configured.
3-DNS	Provides information on the 3-DNS Controllers you have configured.
BIG-IP	Provides information on the BIG-IP systems you have configured.
EDGE-FX	Provides information on the EDGE-FX systems you have configured.
Probers	Provides information on the probers you have configured.
Hosts	This statistics screen provides information on the hosts you have configured.
Virtual servers	Provides information on the virtual servers you have configured.
Weather Map	Provides information on the average round trip times, average completion rates, and average router hops between the data centers or links you have configured and local DNS servers.

Table 6.1 *Configuration utility Statistics screens*

Statistics screen	Description
Paths	Provides information on the paths created by the 3-DNS Controller when paths are required to fulfill name resolution requests.
Local DNS	Provides information on the local DNS servers in the 3-DNS Controller database.

Table 6.1 *Configuration utility Statistics screens*

To view system statistics

1. In the navigation pane, expand the **Statistics** item.
2. From the list, select the item representing the statistics you wish to view.
3. For details about the information displayed on a specific statistics screen, click **Help** on the toolbar.

Overview of the Internet Weather Map

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

- The average round trip time between the local DNS servers on a particular continent and the data centers or links in your network
- The average completion rate between the local DNS servers on a particular continent and the data centers or links in your network
- The average number of router hops between the local DNS servers on a particular continent and the data centers or links 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 2-7.

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.

The round trip time and completion rate data on the Internet Weather Map Statistics screen are based on path metrics. If you do not have path probing activated, the data on this screen is stale. The router hops data are based on

information collected by the **traceroute** utility. If you do not allow the 3-DNS Controller to collect hops information, the average router hops data is stale.

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

1. In the navigation pane, click **System**.
   The System - General screen opens.
2. On the toolbar, click **Metric Collection**.
   The System - Metric Collection screen opens.
3. Check the **Allow Probing** box.
   The 3-DNS Controller can now collect path information for the data centers in your configuration.
4. Check the **Allow Hops** box.
   The 3-DNS Controller 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 or link to each continent
- For each data center or link, 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 or links. 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 or link, 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 or link

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 or link to each continent
- For each data center or link, 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 or over the links. 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 or link, 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 or link, 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 or link

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 or link and each continent
- For each data center or link, 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 or over the links. 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 or link, 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 or link, 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 or link

Interpreting the Internet Weather Map data

You can use the data in the Internet Weather Map (IWM) to compare performance between data centers or links. 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 or link 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 or link best serves content for which continent. By analyzing which data center or link 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.

Working with command line utilities

The 3-DNS includes several command line utilities. These utilities allow you to configure various features of the 3-DNS Controller from the command line. For additional 3-DNS configuration options, you may also want to review the following chapters. For information on working with the Setup utility, see the ***3-DNS Administrator Guide***, Chapter 3, *Using the Setup Utility*.

Viewing command line utilities documentation

You can access the most current documentation on 3-DNS utilities by using the Configuration utility or by using the command line. You can view all the documentation for the 3-DNS Controller from the main screen of the Configuration utility, including the man pages for the utilities that are shipped with the system.

To view 3-DNS man pages using the Configuration utility

1. Log on to the Configuration utility.
2. From the Online Documentation section of the 3-DNS Controller home screen, click **3-DNS Man Pages**.
   A screen containing an index of 3-DNS man pages opens.

To display a list of utilities that fall into a particular category

To display a list of utilities that fall into a particular category, type the following command:

```
man -k <category>
```

For example, to get a list of utilities that pertain to DNS, type the following command, and a list of utilities that pertain to DNS appears.

```
man -k dns
```

To display documentation for a specific 3-DNS utility

To display the man page for a specific utility, type the following command:

```
man <utility>
```

For example, if you type the following command, the **3dparse** man page appears:

```
man 3dparse
```

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 Controller from the command line. You can perform the following tasks:

- Work with security issues
- Work with the **big3d** agent

Figure 6.2 shows the main screen of the 3-DNS Maintenance menu.

```
3-DNS (R)  Maintenance Menu

Configure SSH communication with remote devices
Generate and Copy iQuery Encryption Key
Check remote versions of big3d
Install and Start big3d
Enter 'q' to Quit
```

Figure 6.2 *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.

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 Controller, BIG-IP, or EDGE-FX system that is added to a network. For more information, see *Working with scripts*, on page 6-20.

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 Controller and any BIG-IP systems or EDGE-FX systems in the network. For more information, see *Working with scripts*, on page 6-20.

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 remote versions of big3d

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 systems and EDGE-FX systems known to the 3-DNS Controller.

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 system and EDGE-FX system 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 system and EDGE-FX system in the network.

Working with scripts

The 3-DNS Controller 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 *Working with the 3-DNS Maintenance menu*, on page 6-19.

Note

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

3dns_add script

Use the **3dns_add** script to add a new 3-DNS Controller to an existing sync group in your network. The **3dns_add** script copies all configuration information from an existing 3-DNS Controller onto the new system. For more details on using this script, refer to the ***3-DNS Administrator Guide***, Chapter 10, *Adding a 3-DNS Controller to an Existing Network*.

WARNING

*You can accidentally remove all configuration information on your existing 3-DNS Controller if you do not follow the guidelines in the **3-DNS Administrator Guide,** Chapter 10, **Adding a 3-DNS Controller to an Existing Network**. Use caution when you run this script.*

3dnsmaint script

The **3dnsmaint** script opens the 3-DNS Maintenance menu. See *Working with the 3-DNS Maintenance menu*, on page 6-19, for more information.

3ndc script

The **3ndc** script starts the **3ndc** utility, which is described in the **3ndc** 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 system and EDGE-FX system known to the 3-DNS Controller.

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 and EDGE-FX systems known to the 3-DNS Controller, 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 Controller, BIG-IP system, or EDGE-FX system to a network, you can run the **config_ssh** script, and if no **ssh** key exists on the system, the script configures **ssh** access.

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 Controllers and (if you have any in your network) BIG-IP or EDGE-FX systems. The **install_key** script creates and distributes the iQuery key to all BIG-IP systems, EDGE-FX systems, and other 3-DNS Controllers in your network.

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

```
F5makekey
```

The **F5makekey** program creates the key in the **/usr/local/bin/F5key.dat** directory. The key contains a random length (12-52) of random content (1-255). 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. Once the key is created, you need to move it to the **/config/3dns/etc/F5key.dat** directory. You must then create a link from the **/config/3dns/etc/F5key.dat** directory to the **/usr/local/bin/F5key.dat** directory.

◆ Note

*We recommend that you use the **Generate and Copy iQuery Encryption Key** script to generate the keys that are required for encrypted communications.*

Configuring Email

You can configure the 3-DNS Controller to send email notifications to you, or to other administrators, using the **sendmail** utility. The 3-DNS Controller includes a sample Sendmail configuration file that you can use to start with, but you must customize the Sendmail setup for your network environment before you can use it.

Before you begin setting up Sendmail, you may need to look up the name of the mail exchanger for your domain. If you already know the name of the mail exchanger, refer to *Setting up the sendmail utility*, on page 6-23, for details about setting up the **sendmail** utility itself.

Finding the mail exchanger for your domain

You can use the **nslookup** command on any workstation that is configured for lookup. Once you find the primary IP address for your domain, you can find the mail exchanger for your domain.

To find the mail exchanger for your domain

1. Identify the default server name for your domain. From a workstation capable of name resolution, type the following on the command line:

   ```
   nslookup
   ```

2. The command returns a default server name and corresponding IP address:

   ```
   Default Server: <server name>
   Address:  <server>
   ```

3. Use the domain name to query for the mail exchanger:

   ```
   set q=mx
   <domain name>
   ```

The returned information includes the name of the mail exchanger. For example, the sample information shown in Figure 6.3 lists **bigip.net** as the preferred mail exchanger.

```
bigip.net   preference = 10, mail exchanger =
mail.siterequest.com
bigip.net   nameserver = ns1.bigip.net
bigip.net   nameserver = ns2.bigip.net
bigip.net   internet address = 192.168.112.1
ns1.bigip.net       internet address = 192.168.112.2
ns2.bigip.net       internet address = 192.168.112.3
```

Figure 6.3 *Sample mail exchanger information*

Setting up the sendmail utility

When you actually set up the **sendmail** utility, you need to open and edit a couple of configuration files. Note that the 3-DNS Controller does not accept email messages, and that you can use the **crontab** utility to purge unsent or returned messages, and that you can send those messages to yourself or another administrator.

To set up and start the sendmail utility

1. Copy **/config/sendmail.cf.off** to **/config/sendmail.cf**.
2. To set the name of your mail exchange server, open the **/config/sendmail.cf** and set the **DS** variable to the name of your mail exchanger. The syntax for this entry is:

   ```
   DS<MAILHUB_OR_RELAY>
   ```

3. Save and close the **/config/sendmail.cf** file.
4. If you want to allow the **sendmail** utility to flush outgoing messages from the queue for mail that cannot be delivered immediately:

   a) Open the **/config/crontab** file, and change the last line of the file to read:

   ```
   0,15,30,45 * * * *   root /usr/sbin/sendmail -q >
   /dev/null 2>&1
   ```

   b) Save and close the **/config/crontab** file.
5. If you want to prevent returned or undelivered email from going unnoticed:

   a) Open the **/config/aliases** file and create an entry for **root** to point to you or another administrator at your site:

   ```
   root: networkadmin@SiteOne.com
   ```

   b) Save and close the **/config/aliases** file.

   c) Run the **newaliases** command to generate a new aliases database that incorporates the information you added to the **/config/aliases** file.
6. To turn the **sendmail** utility on, either reboot the system or type the following command:

   ```
   /usr/sbin/sendmail -bd -q30m
   ```

Using a serial terminal with the 3-DNS Controller

There are two ways to add a serial terminal to the 3-DNS Controller. You can add a serial terminal in addition to the console, or you can add a serial terminal as the console. The difference between the two is:

- A serial terminal configured as a terminal displays a simple login. You can log in and run commands and edit files. In this case, you can use the serial terminal in addition to the keyboard and monitor.
- A serial terminal configured as the console displays system messages and warnings in addition to providing a login prompt. In this case, the serial terminal replaces the keyboard and monitor.

To connect the serial terminal to the 3-DNS Controller

Connect a serial line cable between the terminal device and the 3-DNS Controller. On the back of 3-DNS Controller is a male, 9-pin RS232C connector labeled **Terminal**. (Be sure not to confuse this with the fail-over connection which is also a male, 9-pin connector.)

WARNING

Do not use the fail-over cable to connect the serial terminal to the 3-DNS Controller. A null modem cable is required.

The connector is wired as a DTE device, and uses the signals described in Table 6.2.

Pin	Source	Usage
1	External	Carrier detect
2	External	Received data
3	Internal	Transmitted data
4	Internal	Data terminal ready
5	Both	Signal ground
7	Internal	Request to send
8	External	Clear to send

Table 6.2 *Serial line cable signals*

The connector is wired for direct connection to a modem, with receipt of a Carrier Detect signal generating transmission of a login prompt by the 3-DNS Controller. If you are planning to connect to a terminal or to connect

a PC and utilize a terminal emulation program such as HyperTerminal™, you need a null modem cable with the wiring to generate the signals shown in Table 6.2.

◆ Note

*You can achieve acceptable operation by wiring pins 7 to 8 and pins 1 to 4 at the back of the 3-DNS Controller (and turning hardware flow control **off** in your terminal or terminal emulator).*

Configuring a serial terminal in addition to the console

You can configure a serial terminal for the 3-DNS Controller in addition to the standard console.

To configure the serial terminal in addition to the console

1. Connect the serial terminal to the 3-DNS Controller.
2. Configure the serial terminal settings in your terminal or terminal emulator or modem as follows:
   - 9600 baud
   - 8 bits
   - 1 stop bit
   - No parity
3. Open the **/etc/ttys** file and find the line that reads **tty00 off**. Modify it as shown here:

```
# PC COM ports (tty00 is DOS COM1)
tty00 "/usr/libexec/getty default" vt100 in secure
```

4. Save and close the **/etc/ttys** file.
5. Reboot the 3-DNS Controller.

Configuring a serial terminal as the console

You can configure the serial terminal as the console.

To configure the serial terminal as the console

1. Disconnect the keyboard from the 3-DNS Controller.
2. Connect the serial terminal to the 3-DNS Controller. When there is no keyboard connected to the 3-DNS Controller, the 3-DNS Controller defaults to using the serial port for the console.

3. Configure the serial terminal settings in your terminal or terminal emulator or modem as follows:
   - 9600 baud
   - 8 bits
   - 1 stop bit
   - No parity
4. Reboot the 3-DNS Controller.

Forcing a serial terminal to be the console

In the case where you have not yet connected the serial terminal or it is not active when the 3-DNS Controller is turned on, as it might be if you are using a terminal server or dial-up modem, you can force the controller to use the serial terminal as a console. Note that you do not need to disconnect the keyboard if you use this procedure to force the serial line to be the console.

To force a serial terminal to be the console

1. Edit the **/etc/boot.default** file.
   Find the entry **-console auto**. Change this entry to **-console com**.
2. Save the **/etc/boot.default** file and exit the editor.
3. Plug the serial terminal into the serial port on the 3-DNS Controller.
4. Turn on the serial terminal.
5. Reboot the 3-DNS Controller.

WARNING

Once you configure a serial terminal as the console for the 3-DNS Controller, the following conditions apply:

Keyboard/monitor access is disabled, and logging in is only possible using Secure Shell (SSH), if configured, or the serial line.

*If the **/etc/boot.default** file is corrupted, the system does not boot at all. Save a backup copy of the original file and keep a bootable CD-ROM on hand.*

*The **/etc/boot.default** file must contain either the line: **-console com** or the line: **-console auto**. Do not configure both settings. This could cause problems when you attempt to boot the system.*

Shutting down the 3-DNS Controller

When you need to turn the 3-DNS Controller completely off, you need to complete two tasks. The first task is to shut down the 3-DNS software. After you shut down the 3-DNS software, you can turn off the power to the system.

To shut down the 3-DNS software from the command line

1. To shut down the 3-DNS software, type the following command:

   ```
   halt
   ```

2. When you see the following message, it is safe to turn off the power to the physical system:

   ```
   System is halted, hit reset, turn power off, or press
   return to reboot
   ```

WARNING

Do not remove the power supply from the power source to turn off the 3-DNS Controller. Doing so may result in irrevocable damage to the system.

7

Configuring a Redundant System

- Introducing redundant systems
- Synchronizing configurations between units
- Configuring fail-safe settings
- Using network-based fail-over
- Setting a specific 3-DNS Controller as the preferred active unit

Introducing redundant systems

A 3-DNS redundant system consists of two identically configured 3-DNS units, only one of which is active at a given time. The inactive unit serves as a standby which becomes active only in case of failure of the active system, a process called ***fail-over***.

3-DNS redundant systems have special settings that you need to configure, such as VLAN fail-safe settings. One convenient aspect of configuring a redundant system is that once you have configured one of the 3-DNS units, you can simply copy the configuration to the other 3-DNS unit in the redundant system by using the configuration synchronization feature.

There are two basic aspects about working with redundant systems:

- Synchronizing base configurations between two 3-DNS units
- Configuring fail-safe settings for the VLANs

In addition to the simple redundant features available on the 3-DNS Controller, several advanced redundant features are available. Advanced redundant system features provide additional assurance that your content is available if a 3-DNS Controller experiences a problem. These advanced redundant system options include:

- Network-based fail-over
- Setting a specific 3-DNS unit to be the active unit

The attributes you can configure for a redundant system are shown in Table 7.1.

Attributes	Description
Synchronizing configurations	This feature allows you to configure one 3-DNS Controller and then synchronize the configuration with the other 3-DNS Controller.
Fail-safe for VLANs	Fail-safe for VLANs provides the ability to cause a 3-DNS Controller to fail over if it is no longer seeing traffic on a VLAN.
Network-based fail-over	You can configure the 3-DNS Controller to use the network to determine the status of the active unit. You can use network-based fail-over in addition to, or instead of, hard-wired fail-over.
Setting a dominant 3-DNS Controller	You can set up one unit in a pair to be the dominant active 3-DNS Controller. The unit you set up as the dominant 3-DNS Controller always attempts to be active.

Table 7.1 *The attributes you can configure for redundant systems*

Differentiating between a redundant system and a sync group

While a redundant system and a sync group both involve more than one 3-DNS Controller sharing a configuration, they have different purposes. A redundant system has the following general attributes:

- There can only be two units in a redundant system.
- Both units are typically in the same physical location.
- The units use a shared IP address on the network.
- Only one unit is actively accepting and processing name resolutions.
- A redundant system can be a member of a sync group.

A sync group has the following general attributes:

- A sync group can have two or more members.
- Sync group members are usually in multiple data centers.
- Sync group members have unique IP addresses on the network.

Synchronizing configurations between units

Once you complete the initial configuration on the first unit in the system, you can synchronize the configurations between the active unit and the standby unit. When you synchronize a configuration, the following configuration files are copied to the other 3-DNS Controller:

- The common bigdb keys
- Most files in the **/config** directory (except the **bigip_base.conf** file, which contains unique configuration information for the unit)

If you use command line utilities to set configuration options, be sure to save the current configuration to the file before you use the configuration synchronization feature. (Alternately, 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

*The 3-DNS Controller creates a file named **/usr/local/ucs/cs_backup.ucs** prior to installing a configuration file (UCS) from a remote machine.*

To synchronize the configuration from the command line

Synchronize the configuration from the command line using the **bigpipe config sync** command. Use the **bigpipe config sync** command without the **all** option to synchronize only the boot configuration file **/config/bigip.conf**.

The **bigpipe config sync all** command synchronizes the following configuration files:

- The common bigdb keys
- Most files in the **/config** directory (except **bigip_base.conf**)

The **bigpipe config sync running** command synchronizes the running version of **/config/bigip.conf**, which is the image that resides in memory as the system runs. This file is written to memory only on the standby unit, and is therefore not saved.

Configuring fail-safe settings

For maximum reliability, the 3-DNS Controller supports failure detection on both internal and external VLANs. When you arm the fail-safe option on a VLAN, the 3-DNS Controller monitors network traffic going through the VLAN. If the 3-DNS Controller detects a loss of traffic on a VLAN when half of the fail-safe timeout has elapsed, it attempts to generate traffic. The VLAN issues an ARP request for the default route. Any traffic through the VLAN, including a response to the ARP requests, averts a fail-over.

If the active unit does not receive traffic on the VLAN before the timer expires, it initiates a fail-over, switches control to the standby unit, and reboots.

WARNING

You should arm the fail-safe option on a VLAN only after the 3-DNS Controller is in a stable production environment. Otherwise, routine network changes may cause fail-over unnecessarily.

Each interface card installed on the 3-DNS Controller is typically mapped to a different VLAN, which you need to know when you set the fail-safe option on a particular VLAN. You can view VLAN names in the Configuration utility, or you can use the **bigpipe vlan show** command to view VLAN names from the command line.

To arm or disarm fail-safe on a VLAN using the Configuration utility

1. In the navigation pane, click **Network**.
   The VLANs list opens and displays all VLANs.
2. Select a VLAN name.
   The VLAN Properties screen opens.
3. Locate the **ArmFailsafe** check box:
   - To activate VLAN fail-safe, check the **Arm Failsafe** box.
   - To de-activateVLAN fail-safe, clear the **Arm Failsafe** box.
4. If you are activating VLAN fail-safe, in the **Timeout** box, type the maximum time (in seconds) allowed for a loss of network traffic before a fail-over occurs.
5. Click the **Apply** button.

To arm or disarm fail-safe on a VLAN from the command line

To look up the names of the existing VLANs, use the **bigpipe vlan** command with the **show** keyword:

```
b vlan show
```

To arm fail-safe on a particular VLAN, use the **bigpipe vlan** command with the **timeout** and **failsafe arm** keywords:

```
b vlan <vlan_name> timeout <seconds>
b vlan <vlan_name> failsafe arm
```

For example, you have an external VLAN named **vlan1** and an internal VLAN named **vlan2**. To arm the fail-safe option on both VLANs with a timeout of 30 seconds, you need to issue the following commands:

```
b vlan vlan1 timeout 30
b vlan vlan2 timeout 30
b vlan vlan1 failsafe arm
b vlan vlan2 failsafe arm
```

To disarm fail-safe on a particular VLAN, use the **bigpipe vlan** command with the **failsafe disarm** keyword:

```
b vlan <vlan_name> failsafe disarm
```

Save the changes to the configuration with the following command:

```
b save
```

Using network-based fail-over

You can configure a redundant system to use the network to determine the status of the active unit. This is known as ***network-based fail-over***. Network-based fail-over can be used in addition to, or instead of, hard-wired fail-over. You can configure network-based fail-over either using the Configuration utility, or by modifying the bigdb database.

To configure network-based fail-over using the Configuration utility

1. In the navigation pane, click **System**.
   The System - General screen opens.
2. Check the **Redundant Configuration, Network Failover** box.
3. Click the **Update** button.

◆ Note

*You see the **Redundant Configuration** options only if you have a redundant system.*

To configure network-based fail-over in the bigdb database

You can also enable network-based fail-over by changing the settings of specific bigdb keys with the **bigpipe db** command. To enable network-based fail-over from the bigdb database, the **Common.Sys.Failover.Network** key must be set to one (**1**). To set this value to one, type:

```
b db set Common.Sys.Failover.Network=1
b failover init
```

You can use the following keys to lengthen the amount of time that the standby unit waits for a response before resending another ping to its peer (which is the active unit). The default setting is **3** seconds. To change the default setting, use the following **b db** commands:

```
b db set Common.Sys.Failover.NetTimeoutSec=<value>
b failover init
```

You can also specify how many times the standby unit re-issues a ping before the standby unit initiates a fail-over and becomes the active unit. The default setting is **3** retries.To change this frequency, change the value for the following bigdb key using the following **b db** commands:

```
b db set Common.Sys.Failover.NetRetryLimit=<value>
b failover init
```

If you have not configured a floating IP address, which is shared by the units in the redundant system, then you need to set the following bigdb key before you can use network failover.

```
b db Common.FTB.FailoverIp = <ip_address>
b failover init
```

Note

For additional information on the bigdb database and the bigdb keys, refer to Appendix E, ***The bigdb Database and bigdb Keys****. For additional information on the* ***bigpipe*** *command, refer to Appendix C,* ***bigpipe Command Reference****.*

Setting a specific 3-DNS Controller as the preferred active unit

Setting a preferred active unit means overlaying the basic behavior of a unit in a redundant with a preference toward being the active unit. A 3-DNS Controller that is set as the preferred active unit becomes active whenever the two units negotiate for active status.

To clarify how this differs from default behavior, contrast the basic behavior of a redundant system in the following description. Each of the two 3-DNS Controller units in a redundant system has a built-in tendency to try to become the active unit. Each unit attempts to become the active unit at boot time; if you boot two 3-DNS units at the same time, the one that becomes the active unit is the one that boots up first. In a redundant configuration, if the 3-DNS units are not configured with a preference for being the active or standby unit, either unit can become the active unit by becoming active first.

You modify keys in the bigdb database to set a unit as the preferred active or the preferred standby unit in a redundant system. For more information on the bigdb database and keys, refer to Appendix E, *The bigdb Database and bigdb Keys*.

To force a 3-DNS unit to active or standby state

Use the following **bigpipe db** commands to set a 3-DNS unit as the preferred standby unit:

```
b db set Local.Bigip.Failover.ForceStandby = 1
b failover init
```

A 3-DNS unit that is the preferred standby unit can still become the active unit if it does not detect an active unit.

Use the following **bigpipe db** commands to set a 3-DNS unit as the preferred active unit:

```
b db set Local.Bigip.Failover.ForceActive = 1
b failover init
```

A 3-DNS unit that prefers to be active can still serve as the standby unit when it is on a live redundant system that already has an active unit. For example, if an active 3-DNS unit that preferred to be active failed over and

was taken out of service for repair, it could then go back into service as the standby unit until the next time the redundant system needed an active unit, for example, at reboot.

A

3-DNS Configuration File

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

Overview of the 3-DNS configuration file

The 3-DNS configuration file, **wideip.conf**, describes the components in a global network, and the services that are available on those components. 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 Controller, 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 **box** statement
- At least one **server** statement defining a 3-DNS Controller
- At least one virtual server, which is defined as part of a **server** statement for one of these objects: BIG-IP system, EDGE-FX system, or host
- 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 Controller reverts to standard DNS behavior.

To edit the 3-DNS configuration file, wideip.conf

1. To ensure that the configuration files contain the same information as the memory cache, type the following command:

   ```
   3ndc dumpdb
   ```

2. Open the **wideip.conf** file in a text editor (either **vi** or **pico**).
3. Make any changes to the configuration that you want to make.
4. Save and close the file.
5. Commit the changes to the configuration by typing:

   ```
   3ndc reload
   ```

Using include files

Include files are files that contain configuration information about one aspect of your network, and are listed in the **wideip.conf** configuration file. For example, you can have one **include** file that defines the topology records for load balancing, and another **include** file that contains local DNS server and path information.

Using **include** files reduces the size of the **wideip.conf** file and makes it easier to manage your configuration. The 3-DNS Controller 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 syntax shown in Figure A.1 when incorporating **include** files into a **wideip.conf** file.

```
include root_in "/config/3dns/include"
include root_out "/config/3dns/include"
include topology "topology.inc"
include geoloc "netIana.inc"
include ldns "ldns.inc"
include region "region.inc"
```

Figure A.1 *Syntax for include files*

Definitions of include statements

Table A.1 lists the **include** statements in the **wideip.conf** file, their descriptions, and their default file names.

Parameter	Description	Default file name
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 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.	**netIana.inc**

Table A.1 *Include file descriptions*

Parameter	Description	Default file name
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**

Table A.1 *Include file descriptions*

Working with statements

A top-level statement in the **wideip.conf** file 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 Controller.
- **globals**
  The **globals** statement defines system-level settings for any 3-DNS Controller configuration options, and sets the defaults for other statements.
- **datacenter**
  The **datacenter** statement defines the group of 3-DNS Controllers, BIG-IP systems, EDGE-FX systems, routers, and hosts that reside in a single physical location.
- **box**
  The **box** statement defines the attributes of the physical device that the 3-DNS software, BIG-IP software, EDGE-FX software, router, or host runs on.
- **server**
  The **server** statement defines any of the following objects: a 3-DNS Controller, a BIG-IP system, an EDGE-FX system, a router, or a host.
- **sync_group**
  The **sync_group** statement defines the group of 3-DNS Controllers and Link Controllers that synchronize their configuration settings and metrics data.
- **wideip**
  The **wideip** statement defines a wide IP and the pool or pools that are used for load balancing. A ***wide IP*** maps a domain name (for example, **stocks.siterequest.com**) to a load balancing mode and a set of virtual servers that serve the content for the domain.

- **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** statement
  - **include** statement
  - **datacenter** statement
  - **box** statement
  - **server** statement
  - **sync_group** statement
  - **wideip** statement
- **Address and port specification**
  For virtual servers, 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_address>:<port>

address <ip_address>
port <port>

address <ip_address>
service <"http">
```

Note

This chapter contains several syntax examples. For an overview of the typography conventions in the syntax examples, see ***Identifying command syntax****, on page 1-2.*

The globals statement

The **globals** statement contains the global options that are used by the 3-DNS Controller, 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 you use a **globals** sub-statement more than once, the 3-DNS Controller uses the last listed value, and does not generate an error message. For example, if your **globals** statement contains the parameters shown in Figure A.2, the 3-DNS Controller uses the value **50** for the time-to-live for metrics collected from a BIG-IP system.

```
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 to include only those sub-statements that you want to change from the default.

```
globals {
  [ time_tolerance <number> ]
  [ autosync < yes | no > ]
  [ autoconf <yes | no> ]
  [ 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_get_link_data <number> ]
  [ timer_get_link_status <number> ]
  [ timer_get_autoconfig_data <number> ]
  [ timer_check_keep_alive <number> ]
  [ timer_persist_cache <number> ]
  [ timer_sync_state <number> ]
  [ dc_prefix <string> ]
  [ 3dns_ttl <number> ]
  [ bigip_ttl <number> ]
  [ edgefx_ttl <number> ]
  [ host_ttl <number> ]
  [ vs_ttl <number> ]
  [ path_ttl <number> ]
  [ trace_ttl <number> ]
  [ link_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_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_coeff_lcs <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>
  [ qos_factor_lcs <number>
  [ default_alternate < ga | null | random | ratio | static_persist |
    packet_rate | leastconn | return_to_dns | rr | topology | vs_capacity
    | kbps | drop_packet | explicit_ip > ]
  [ default_fallback < completion_rate | ga | hops | leastconn |
    null | packet_rate | qos | random | ratio | return_to_dns |
    rr | rtt | topology | vs_capacity | static_persist | kbps | drop_packet |
explicit_ip > ]
  [ fb_respect_depends < yes | no > ]
  [ fb_respect_acl < yes | no > ]
  [ aol_aware < yes | no >
  [ path_duration <number> ]
  [ ldns_duration <number> ]
  [ prober <ip_address> ]
  [ 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 > ]
  [ probe_protocol {
      [ dns_rev ]
      [ dns_dot ]
      [ udp ]
      [ tcp ]
      [ icmp ]
    } ]
  [ datasize_system  <number> ]
  [ datasize_reap_pct  <number> ]
  [ default_iquery_protocol < udp | tcp > ]
  [ traceroute_port <number> ]
}
```

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

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

```
globals {
  autoconf yes              //Auto-discovery is available for servers
  autosync yes              //Synchronization is on
  time_tolerance 10         //Ten seconds is maximum variance for 3-DNS Controllers times
  prober 192.168.101.2      // Default prober is New York 3-DNS Controller
  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** statement can contain some or all of the parameters that are described in the following sections.

Auto-discovery

The auto-discovery sub-statement, **autoconf**, specifies whether the 3-DNS Controller can gather virtual server information for any servers you define. You can also specify, for individual servers, whether auto-discovery is on. (See *Auto-discovery for servers*, on page A-30, for more information.) If you disable auto-discovery in the **globals** statement, you override the **autoconf** setting in the **server** statements.

Parameter	Description	Default
autoconf	Specifies whether the auto-discovery option is available for servers. Note that setting this global variable to **no** disables auto-discovery for any servers that you may configure.	yes

Table A.2 *Auto-discovery sub-statement*

◆ Note

In the Configuration utility, the list of self IPs reported by auto-discovery may not include all self IPs on the system. Auto-discovery reports only public or external addresses, and any associated address translation will not show up in the Configuration utility's self-IP list.

◆ Note

In the Configuration utility, auto-discovery is labeled ***Discovery****.*

Synchronization

The synchronization sub-statements specify how the current 3-DNS Controller handles synchronizing its configuration with the other 3-DNS Controllers in the network.

Parameter	Description	Default
autosync	When **autosync** is **on**, the 3-DNS Controller compares file time stamps among the 3-DNS Controllers in your network. When a controller has a file that has a more recent time stamp than the same file on the other 3-DNS Controllers in the network, the controller with the most current file synchronizes that file to the other controllers.	yes
time_tolerance	Specifies the variation of time allowed (in seconds) when comparing time stamps on files. File time stamps are included in iQuery messages when they are passed among the 3-DNS Controllers in the network. When a file's time stamp is more than **time_tolerance** seconds ahead of the time stamp for the same file on another controller, the 3-DNS Controller updates the older file's time stamp with the more recent time stamp. If the difference between the two time stamps falls within the **time_tolerance** setting, the controller considers the files to be the same and does not overwrite one with the other. Set the **time_tolerance** global to **0** if you want to disable it.	10
sync_gui_rules	Specifies whether the 3-DNS Controller synchronizes production rules with the other 3-DNS Controllers in the network.	no
sync_named_conf	Specifies whether the 3-DNS Controller synchronizes the DNS zone files with the other 3-DNS Controllers in the network.	yes
sync_timeout	Specifies the amount of time the controller waits for the synchronization to complete, after the synchronization process for configuration data starts. If the synchronization process does not complete in this time, the controller stops the process, logs an error, and retries the synchronization at the next interval.	180
sync_zones_timeout	Specifies the amount of time the controller waits for the synchronization to complete, after the synchronization process for zone files starts. If the synchronization process does not complete in this time, the controller stops the process, logs an error, and retries the synchronization at the next interval.	300

Table A.3 *Synchronization sub-statement*

Encryption

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

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

Table A.4 *Encryption sub-statements*

Dependencies

The dependencies sub-statements specify whether the 3-DNS Controller 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 that the 3-DNS Controller checks the availability of virtual servers on BIG-IP systems, EDGE-FX systems, and hosts before it uses the virtual servers for load balancing. 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 Controller 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.5 *Dependencies sub-statement*

LDNS persistence

Dynamic load balancing modes and wide IP persistence depend on path information to resolve requests. The value for **persist_ldns** must be set to **yes** (the default) so that the 3-DNS Controller stores and uses path information. If you use only static load balancing modes and are not using persistence, you can set **persist_ldns** to **no** to conserve memory.

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

Table A.6 *LDNS persistence sub-statement*

Load balancing persistence

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

Parameter	Description	Default
default_persist_ttl	Specifies the length of time the 3-DNS Controller retains persistent connections information before the information is purged.	3600

Table A.7 *Load balancing persistence sub-statements*

Parameter	Description	Default
persist_mask	Specifies the significant bits of an LDNS IP address to use with the **static_persist** load balancing mode. The default setting indicates that all 32 bits are significant.	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.30, on page A-38, for more information.	yes

Table A.7 *Load balancing persistence sub-statements*

Periodic task intervals

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

Parameter	Description	Default
timer_get_3dns_data	Specifies how often the 3-DNS Controller retrieves availability data for other 3-DNS Controllers in the network. You can enter a value between **1** and **3600** seconds.	20
timer_get_server_data	Specifies how often the 3-DNS Controller refreshes 3-DNS Controller, BIG-IP system, and EDGE-FX system information. You can enter a value between **1** and **3600** seconds.	20
timer_get_host_data	Specifies how often the 3-DNS Controller refreshes other host machine information. You can enter a value between **1** and **3600** seconds.	90
timer_get_vs_data	Specifies how often the 3-DNS Controller refreshes virtual server information. You can enter a value between **1** and **3600** seconds.	30
timer_get_path_data	Specifies the minimum interval at which the 3-DNS Controller sends requests for new or updated path information (for example, round trip time or ping packet completion rate) to the **big3d** agents in the network. Note that if the controller is monitoring a large number of paths, the controller sends the requests to the **big3d** agents more frequently. You can enter a value between **1** and **600** seconds.	120
timer_get_ecv_data	Specifies how often the 3-DNS Controller refreshes ECV information. You can enter a value between **5** and **3600** seconds.	90
timer_get_trace_data	Specifies how often the 3-DNS Controller 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 **3600** seconds.	60
timer_get_autoconfig_data	Specifies how often the 3-DNS Controller checks for updated virtual server configuration information, when auto-discovery is enabled.	30

Table A.8 *Periodic task interval sub-statements*

Parameter	Description	Default
timer_get_link_data	Specifies how often the 3-DNS Controller refreshes link metrics information.	10
timer_get_link_status	Specifies how often the 3-DNS Controller refreshes link availability information.	2
timer_check_keep_alive	Specifies how often the 3-DNS Controller queries remote 3-DNS Controllers, BIG-IP systems, and EDGE-FX 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 **3600** seconds.	60
timer_persist_cache	Specifies how often the 3-DNS Controller writes the **wideip.conf** file from memory. You can enter a value between **1** and **604800** seconds.	3600

Table A.8 *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 Controller refreshes the information.

Parameter	Description	Default
default_ttl	Specifies the default number of seconds that the 3-DNS Controller considers a 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 Controller considers performance data for the other 3-DNS Controllers to be valid.	60
bigip_ttl	Specifies the number of seconds that the 3-DNS Controller can use BIG-IP system information for name resolution and load balancing. You can enter a value between **1** and **10800**. The following relationship should be maintained: **bigip_ttl** is greater than **timer_get_server_data**. A 3:1 ratio is the optimal setting for this relationship.	60
edgefx_ttl	Specifies the number of seconds that the 3-DNS Controller can use EDGE-FX system information for name resolution and load balancing. You can enter a value between **1** and **10800**. The following relationship should be maintained: **edge_ttl** is greater than **timer_get_server_data**. A 3:1 ratio is the optimal setting for this relationship.	60
host_ttl	Specifies the number of seconds that the 3-DNS Controller can use host information for name resolution and load balancing. You can enter a value between **1** and **10800**. The following relationship should be maintained: **host_ttl** is greater than **timer_get_host_data**.	240

Table A.9 *Data time-outs sub-statements*

Parameter	Description	Default
vs_ttl	Specifies the number of seconds that the 3-DNS Controller can use virtual server information (data acquired about a virtual server from a BIG-IP system, an EDGE-FX system, or host) for name resolution and load balancing. You can enter a value between **1** and **10800**. The following relationship should be maintained: **vs_ttl** is greater than **timer_get_vs_data**.	120
link_ttl	Specifies the number of seconds that the 3-DNS Controller can use link information for link load balancing. You can enter a value between **1** and **10800**. The following relationship should be maintained: **link_ttl** is greater than **timer_get_link_data**.	31
path_ttl	Specifies the number of seconds that the 3-DNS Controller can use path information for name resolution and load balancing. You can enter a value between **1** and **2419200**. The following relationship should be maintained: **path_ttl** is greater than **timer_get_path_data**.	2400
trace_ttl	Specifies the amount of time (in seconds) that the 3-DNS Controller considers **traceroute** data, for router hops, to be valid. You can enter a value between **1** and **2419200**.	604800 (seven days)

Table A.9 *Data time-outs sub-statements*

Probing and metrics collection

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

Parameter	Description	Default
rtt_timeout	Specifies, in seconds, how long the **big3d** agent waits for a response to a probe. You can enter a value between **1** and **10**.	3
rtt_sample_count	Specifies the number of packets to send from the **big3d** agent to the LDNS, host, or host virtual server to determine the path information between those two systems. You can type a value between **1** and **10**.	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
probe_protocol	Determines which protocols the 3-DNS Controller uses to probe LDNS servers to calculate path round trip times, and in what order the protocols are used. You can specify one or more of the following protocols: **icmp**, **udp**, **tcp, dns_dot,** and **dns_rev**.	icmp
default_probe_limit	Specifies the number of times that the 3-DNS Controller attempts to probe a failed path before continuing.	12
paths_never_die	Specifies that the 3-DNS Controller should not attempt to refresh path data for a path that has been successfully probed.	no

Table A.10 *Probing and metrics collection sub-statements*

Parameter	Description	Default
check_dynamic_depends	Specifies that the 3-DNS Controller 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_probe	Specifies that the 3-DNS Controller 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 Controller should collect hops metrics when probing paths.	yes
prober	Specifies the default prober to perform availability service checks on hosts and host virtual servers. When this option is set to **127.0.0.1**, the controller chooses the best **big3d** agent to probe the host or host virtual server. The best **big3d** agent is defined as the agent that is closest to the host (for example, a **big3d** agent on the same link is closer than in the same data center, which is closer than any **big3d** agent in the network), and has the least number of outstanding probe requests compared to the agent's probe capacity. You can override this default setting within the **server** statement. Note that this sub-statement is optional if the 3-DNS Controller does not manage hosts or host virtual servers.	127.0.0.1

Table A.10 *Probing and metrics collection sub-statements*

Resource limits

The resource limits sub-statements define the amount of memory on the 3-DNS Controller 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 Controller can use when receiving iQuery data. You can enter a value between **8192** and **262144**.	262144
tx_buf_size	Specifies the maximum amount of socket buffer data memory the 3-DNS Controller can use when transmitting iQuery data. You can enter a value between **8192** and **262144**.	262144

Table A.11 *Resource limits sub-statements*

Topology settings

The Topology load balancing mode uses geographic data to determine the best virtual server to send in response to a request. The topology globals affect how the 3-DNS Controller uses the **topology** and **regions include** files. For more information about include files, see *Using include files*, on page A-2. For more information about the Topology load balancing mode, see Chapter 3, *Topology*.

Parameter	Description	Default
dump_topology	Specifies whether the 3-DNS Controller writes out the topology statement whenever a configuration change occurs.	yes

Table A.12 *Topology sub-statement*

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 Controller 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)] +
[(qos_coeff_lcs) * (score_lcs)]
```

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
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_coeff_lcs	Specifies the relative weighting for link capacity when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	30
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

Table A.13 *QOS values sub-statements*

Parameter	Description	Default
qos_factor_kbps	Specifies the factor used to normalize raw kilobytes per second values when computing the QOS score.	1000
qos_factor_lcs	Specifies the factor used to normalize raw link capacity values when computing the QOS score.	1

Table A.13 *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 Controller respects virtual server status when load balancing switches to the specified fallback mode.	no
fb_respect_acl	Determines whether the 3-DNS Controller imposes topology access control when load balancing switches to the specified fallback mode.	no
aol_aware	Determines whether the 3-DNS Controller recognizes local DNS servers that belong to the Internet service provider, America Online (AOL).	yes

Table A.14 *Load balancing sub-statements*

Buffer size

The buffer size sub-statements specify the maximum amount of UDP data that the 3-DNS Controller 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.	262144
resolver_tx_buf_size	Specifies the wide IP **send** buffer size.	262144

Table A.15 *Buffer size sub-statements*

Reaping

The 3-DNS Controller 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 Controller. ***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 Controller reserves for system usage, such as non-3-DNS specific processes. Note that the system determines the value for this variable.	varies by system
datasize_reap_pct	Specifies what percentage of memory that the 3-DNS Controller 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 **31536000** (one year).	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 **31536000** (one year).	2419200 (28 days)

Table A.16 *Reaping sub-statements*

iQuery port options

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

Parameter	Description	Default
use_alternate_iq_port	Determines whether the 3-DNS Controller 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 Controller 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 iQuery traffic.	yes

Table A.17 *iQuery port options sub-statements*

The datacenter statement

A **datacenter** statement defines the group of 3-DNS Controllers, BIG-IP systems, EDGE-FX systems, routers, 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"> ]
[ router <ip_address | "name"> ]
[ host <ip_address | "name"> ]
}
```

Figure A.6 *Syntax for the **datacenter** statement*

Figure A.7 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
  router 192.168.101.254
  host 192.168.105.40
}
```

Figure A.7 *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. This parameter must be enclosed in quotation marks.
location	Specifies the location of the data center. This parameter 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.

Table A.18 *Data center sub-statements*

Parameter	Description
contact	Identifies the administrator of the data center. This parameter 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 or name of a 3-DNS Controller in this data center. If you have a redundant system, you need to specify only one controller's IP address.
bigip	Specifies the IP address or name of a BIG-IP system in this data center. If you have a redundant system, you need to specify only one controller's IP address.
edgefx	Specifies the IP address or name of an EDGE-FX system in this data center.
router	Specifies the IP address or name of a router in this data center.
host	Specifies the IP address or name of a host in this data center.

Table A.18 *Data center sub-statements*

The box statement

The **box** statement defines the name, IP addresses, remote user, **big3d** factories, and iQuery protocol associated with an actual physical system. The physical system can have one or more servers running on it. For example, if you have a BIG-IP system running the 3-DNS Controller module, you define one physical system (a box), and two server types (the 3-DNS Controller and the BIG-IP system) in the **wideip.conf** file.

Syntax for the box statement

Figure A.8 shows the syntax for a **box** statement.

```
box {
  [ name <"box name"> ]
    address <ip_address> [ translates to <ip_address> ]
  [ address <ip_address> [ translates to <ip_address> ]]...
    iquery_protocol < udp | tcp >
  [ remote {
     [ user <"user_name"> ]
     [ secure <yes | no> ]
    } ]
  [ factories {
     [ prober <number> ]
     [ hops <number> ]
     [ ecv <number> ]
     [ snmp <number> ]
    } ]
}
```

Figure A.8 *Example syntax for the* ***box*** *statement*

Definitions of box sub-statements

The **box** statement can contain some or all of the parameters that are described in the following sections.

Address information

The address information sub-statements provide the basic attributes of a box. Note that if you are defining a 3-DNS Controller, a BIG-IP system, an Link Controller, or an EDGE-FX system, you can define the iQuery protocol and factories for the **big3d** agent, in addition to the name and IP addresses for the box.

Parameter	Description
name	Specifies the name of the physical unit that the server type runs on. You must enclose all names in quotation marks. This is an optional setting.
address	Specifies the public IP address of the interface in the physical unit. Note that you can have several addresses for one box.
translates to translates_to	Specifies the private IP address the physical unit. If you have firewalls in your network, the **translates_to** address is the IP address that is on your internal network. This is an optional parameter for the **address** parameter, and you can use either syntax format.
iquery_protocol	Specifies the iQuery transport option, **tcp** or **udp**. Applies to 3-DNS Controllers, BIG-IP systems, and EDGE-FX systems only.

Table A.19 *Address information sub-statements for the* ***box*** *statement*

Remote connections

You use the **remote** sub-statement only if you want to specify a different login name for a 3-DNS Controller, BIG-IP system, or EDGE-FX system.

Parameter	Description
remote	Indicates the start of a **remote** sub-statement. Applies to 3-DNS Controllers, BIG-IP systems, and EDGE-FX systems only.
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 3-DNS Controllers, BIG-IP systems, and EDGE-FX systems only.
secure	Specifies whether the communications with the remote system are encrypted. The default setting is **yes**. We recommend that you do not change this setting.

Table A.20 *Remote connections sub-statements*

Factories

For any box that is running a 3-DNS Controller, BIG-IP system, or EDGE-FX 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 Controller uses the default settings. For more information on factories and probing, see *Working with the big3d agent*, on page 5-1. For information on configuring probing and metrics collection in the **wideip.conf** file, see *Probing and metrics collection*, on page A-13.

Parameter	Description
factories	Indicates the start of the factories definition. Applies to 3-DNS Controller, BIG-IP systems, and EDGE-FX 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 **0**.
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.21 *Factories sub-statements*

Note

*In previous releases, you configured the **factories** sub-statement within a **server** statement. With this release, you now define factories within a **box** statement. Note that is still valid to define factories in a **server** statement, but we recommend that you define factories in the **box** statement.*

The server statement

The **server** statement defines the characteristics associated with a particular 3-DNS Controller, BIG-IP system, EDGE-FX system, host, or router. A basic **server** statement contains the following information:

- The type of server: 3-DNS Controller, BIG-IP system, EDGE-FX system, router, or host
- One of the IP addresses, or the name, of the physical unit that the server is running on
- If the server is a BIG-IP system, EDGE-FX Cache, or host, the set of virtual servers that is available on it
- If the server is a router, the link or links managed by the router

You can also define several parameters that are applicable to the specific server type. As such, the syntax and examples for each server type are listed separately in the following sections of this chapter. All sub-statements for the **server** statement are defined in the *Definition of server sub-statements* section, which starts on page A-29.

Syntax for the server statement (3-DNS Controller)

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

```
server {
  type 3dns
  name <"3dns_name">
  box <ip_address | "name">
[ box <ip_address | "name"> //use for a redundant system
[ autoconf <disable | enable_with_auto_delete_disabled | enable> ]
[ link_autoconf <disable | enable_with_auto_delete_disabled | enable> ]
[ disabled_until [<time in seconds> [<by_whom>]]]
}
```

***Figure A.9 Server** statement syntax for defining a 3-DNS Controller*

Figure A.10 shows an example of a 3-DNS Controller definition in the **wideip.conf** file.

```
// 3-DNS Controller in New York data center
server {
   type 3dns
   name "3dns-newyork"
   box 192.168.101.2
}
```

***Figure A.10** Example of a 3-DNS Controller definition*

Syntax for the server statement (BIG-IP system)

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

```
server {
  type bigip
[ name <"bigip_name"> ]
  box <ip_address | "name">
[ box <ip_address | "name"> //use for a redundant system
[ autoconf <disable | enable_with_auto_delete_disabled | enable> ]
[ link_autoconf <disable | enable_with_auto_delete_disabled | enable> ]
[ iquery_protocol [udp | tcp] ]
[ prober <ip_address> ]
[ disabled_until [<time in seconds> [<by_whom>]]]
[ limit {
      [ kbytes_per_sec <number> ]
      [ pkts_per_sec <number> ]
      [ current_conns <number ]
      } ]
  vs {
     address <ip_address>:<port number>
   [ translates to <ip_address:port> ]
   [ 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
     } ]
   [ disabled_until [<time in seconds> [<by_whom>]]]
  }
[ vs address <ip_address>:<port number> [ translates to <ip_address:port> ]
}
```

***Figure A.11 Server** statement syntax for defining a BIG-IP system*

Figure A.12 shows an example definition of a BIG-IP non-redundant system that is behind a firewall, and that has two virtual servers.

```
server {
   type            bigip
   box             192.168.101.40
   name            "bigip-newyork"
   iquery_protocol  udp
   autoconf enable
   vs  192.168.101.50:80 translates to 10.0.0.50:80
   vs  192.168.101.50:25 translates to 10.0.0.50:25
}
```

***Figure A.12** Example syntax for defining a BIG-IP system*

Syntax for the server statement (EDGE-FX system)

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

```
server {
  type edgefx
[ name <"edgefx_name"> ]
  box <ip_address | "name">
[ autoconf <disable | enable_with_auto_delete_disabled | enable> ]
[ disabled_until [<time in seconds> [<by_whom>]]]
[ 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 ] ]
  [ snmp  {  //required
       agent edgefx
       version 2
       community <"public">
    } ]
    vs {       //applicable only to EDGE-FX Cache
      address <ip_address>:<port number>
    [ translates to <ip_address:port> ]
    [ ratio <number> ]
    [ depends_on {
        <ip_address>:<port number>
      [ <ip_address>:<port number> ]
      }
    [ limit {
      [ cpu_avail <number> ]
      [ disk_avail <number> ]
      [ mem_avail <number> ]
      [ kbytes_per_sec <number> ]
      [ pkts_per_sec <number> ]
      [ current_conns <number> ]
      } ]
    [ disabled_until [<time in seconds> [<by_whom>]]]
    }
  [ vs address <ip_address>:<port number> [ translates to <ip_address:port> ]]
}
```

Figure A.13 *Example syntax for defining an EDGE-FX system*

◆ Note

An EDGE-FX system can be either an EDGE-FX Cache or a GLOBAL-SITE Controller, however, only an EDGE-FX Cache can manage virtual servers.

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 *Working with SNMP on the 3-DNS Controller*, on page 5-12.

```
server {
  type host
[ name <"host_name"> ]
  box <ip_address | "name">
[ probe_protocol <tcp | icmp | dns_rev | dns_dot> ]
[ prober <ip_address> ]
[ port <port number> | service <"service name"> ]
[ autoconf <disable | enable_with_auto_delete_disabled | enable> ]
[ snmp {
    agent <generic | ucd | solstice | ntserv | win2kserv | ciscold | ciscold2 | ciscold3
| foundry | arrowpoint | alteon | cacheflow | netapp>
    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> ]
  } ]
[ disabled_until [<time in seconds> [<by_whom>]]]
  vs {
    address <ip_address>:<port number>
  [ translates to  <ip_address>:<port number> ]
  [ 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
    } ]
  [ disabled_until [<time in seconds> [<by_whom>]]]
  }
[ vs address <ip_address>:<port number> [ translates to  <ip_address>:<port number> ]
}
```

***Figure A.14 Server** statement syntax for defining a host*

Figure A.15 shows an example definition of a load balancing host with two virtual servers.

```
server {
   type            host
   name            "host-tokyo"
   box             192.168.104.40
   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.15 *Example syntax for defining a host*

Syntax for the server statement (router)

The following **server** statement syntax applies to routers only. When you define a router, you also define the router's links.

```
server {
   type            router
 [ name            "router_name" }
   box <ip_address | "name">
 [ port <port number> ]
   snmp {
      agent router
      version <number>
      community "public"
   }
   link {
      name      "link_name"
      address   <ip_address>
    [ address   <ip_address> ]
    [ uplink_address <ip_address> ] //use only with SNMP
    [ isp       <"isp_name"> ]
    [ duplex <yes | no> ]
    [ cost {
          ratio   <number>
        [ prepaid <number>Kb ]
        [ rate {
            //Up to      $/Mb/Sec
            <number>Kb    <number>
          [ <number>Kb    <number> ]
          } ]
      } ]
    [ in {
          limit {
            kbytes_per_sec <number>
          }
      } ]
    [ out {
          limit {
            kbytes_per_sec <number>
          }
      } ]
    [ total {
          limit {
            kbytes_per_sec <number>
          }
      } ]
   }
  [ link {
       ...
    } ]
}
```

***Figure A.16 Server** statement syntax for defining a router*

Figure A.17 shows an example definition of a router with one link.

```
server {
   type            router
   name            "my_router"
   box             "my_router"
   port            0
   duplex          yes
   link {
      name      "Link_1"
      address   192.168.104.50
      isp       "My_ISP"
      cost {
          ratio   1
          prepaid      0Kb
      }
   }
}
```

Figure A.17 *Example syntax for defining a router*

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.22 lists the parameters of the address information sub-statement.

Parameter	Description
type	Indicates whether the specified server is a 3-DNS Controller, BIG-IP system, EDGE-FX system, router, or host.
box	Specifies the IP address of the 3-DNS Controller, BIG-IP system, EDGE-FX system, router, or host. If you have a redundant system, you specify two box IP addresses, one for each unit. If you are defining a multi-homed system, then you specify only one IP address using this parameter.
name	Specifies the name of the 3-DNS Controller, BIG-IP system, EDGE-FX system, router, or host. You must enclose all names in quotation marks.
probe_protocol	Specifies the protocol method to use for probing: **icmp**, **tcp**, **dns_rev**, or **dns_dot**. Applies to hosts only.

Table A.22 *Address information sub-statements*

Parameter	Description
prober	Specifies the IP address of the system probing the host or router. This IP address points to a BIG-IP system, a 3-DNS Controller, or an EDGE-FX system that runs the **big3d** agent. If you omit this parameter, the 3-DNS Controller uses the **prober <ip_address>** parameter defined in the **globals** statement. The **big3d** agent probes the host and virtual servers to verify whether the host or a particular virtual server is currently available to accept connections. Note that if you are configuring a prober for a router, the prober system (that is, the BIG-IP system or the 3-DNS Controller) must be in the same data center as the router. This parameter applies to hosts and routers only.
port	Specifies the port used to probe the host or router if the **probe_protocol** parameter is set to TCP. This applies to hosts and routers only.

Table A.22 *Address information sub-statements*

Auto-discovery for servers

Once you have added the address information for a server, you can enable the auto-discovery option (**autoconf**), and the 3-DNS Controller automatically gathers virtual server information (for servers). You can also enable the link_autoconf option to automatically gather link and router information for the server's data center. Note that if you have disabled the global variable **autoconf**, then these parameters in the **server** statement is inoperative. (See *Auto-discovery*, on page A-8, for more information.)

Parameter	Description
autoconf	Specifies whether the 3-DNS Controller initially gathers, and then maintains, the virtual server information for the server. There are three settings for this parameter: **disable**, **enable_with_auto_delete_disabled**, and **enable**. The default setting is **disable**.
link_autoconf	Specifies whether the 3-DNS Controller initially gathers, and then maintains, the router and link information for the server's data center. There are three settings for this parameter: **disable**, **enable_with_auto_delete_disabled**, and **enable**. The default setting is **disable**. Note that if you do not enable the **autoconf** parameter, the **link_autoconf** parameter remains disabled regardless of the setting you specify.

Table A.23 *Auto-discovery sub-statements*

◆ Note

In the Configuration utility, the list of self IPs reported by auto-discovery may not include all self IPs on the system. Auto-discovery reports only public or external addresses, and any associated address translation does not show up in the Configuration utility's self-IP list.

◆ Note

In the Configuration utility, the auto-discovery options are labeled ***Discovery*** *and* ***Link Discovery****.*

Limit settings

Using the **limit** sub-statement, you can manage the physical and throughput resources of your BIG-IP systems, EDGE-FX systems, hosts, and their respective virtual servers. If you omit this sub-statement, the 3-DNS Controller does not use resource thresholds to monitor the availability of the BIG-IP systems, EDGE-FX systems, 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 systems 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 systems 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 systems 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 systems 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.24 Limit** sub-statement*

SNMP settings

The **snmp** sub-statement is valid for hosts, EDGE-FX Caches, and routers 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 a host or router, refer to the documentation provided with that system.

Parameter	Description
snmp	Specifies the start of an SNMP definition.
agent	Specifies the SNMP agent type. If you omit this parameter for hosts, the **big3d** agent uses the generic SNMP agent.
port	Specifies the port the SNMP agent runs on.

***Table A.25** SNMP sub-statements*

Parameter	Description
community	Specifies the password for basic SNMP security and for grouping SNMP hosts. Enclose this string in quotation marks.
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.
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 and routers only.
version	Specifies the SNMP agent version. Applies to hosts and routers only.

Table A.25 *SNMP sub-statements*

Virtual server definitions

Part of defining a BIG-IP system, EDGE-FX system, or host server type 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	Specifies the virtual server's port number. You can add the port number, preceded by a colon, on the same line as the virtual server's address. You can also 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 Controller uses this virtual server for load balancing.
probe_protocol	Specifies the protocol to use for probing this virtual server: ICMP or TCP.
translates to	Specifies that iQuery packets sent to the big3d agent 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.

Table A.26 *Sub-statements for virtual server definitions*

Link definitions

When you define a router in the configuration, you also define at least one link for the router. Table A.27 lists the available parameters for a link definition.

Parameter	Definition
link	Indicates the start of a link definition.
name	Specifies the name of the link. Names must be enclosed in quotation marks.
address	Specifies the IP address associated with the link.
isp	Specifies the name of the Internet service provider (ISP) associated with the link. Note that this is an optional parameter.
duplex	Specifies whether the ISP uses the duplex billing method for your bandwidth usage. The default setting is **yes**.
uplink_address	Specifies the IP address on the router that is associated with the ISP. Note that the uplink IP address is used for SNMP metrics gathering only.
cost	Indicates the start of the link load balancing parameters.
ratio	Specifies the volume of link traffic that should use this link in comparison to other configured links. Note that links are load balanced either by using the ratio parameter, or by using the prepaid and incremental parameters.
prepaid	Specifies, in kilobits (Kb), the amount of bandwidth that is paid for each month, regardless of usage.
rate	Specifies, in kilobits (Kb), a segment of bandwidth and its associated cost.
in	Specifies a limit, in kilobits, on the inbound traffic on the link.
out	Specifies a limit, in kilobits, on the outbound traffic on the link.
total	Specifies a limit, in kilobits, on the total of inbound and outbound traffic on the link.

Table A.27 *Sub-statements for link definitions*

The sync_group statement

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

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.18 *Syntax for the **sync_group** statement*

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

Figure A.19 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.19 *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 name of a 3-DNS Controller in the group. First list the IP address of the principal system. Then list all other 3-DNS Controllers, in the order that they should become a principal system, if the previously listed principal 3-DNS Controller fails. Note that there can only be one principal system in a sync group at any time. If you are adding a redundant system to a sync group, you can specify any of the IP addresses for either unit in the **sync_group** statement.

Table A.28 ***Sync_group** sub-statements*

The wideip 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">
  name <"domain_name">
[ alias <"alias_name"> ... ]
[ ttl <number> ]
[ persist < yes | no > ]
[ persist_ttl <number> ]
[ port_list <port_number> <port_number> ... ]
[ manual_resume < yes | no > ]
[ disabled_until [<time in seconds> [<by_whom>]]]
[ qos_coeff {
    rtt <number>
    hops <number>
    completion_rate <number>
    packet_rate <number>
    vs_capacity <number>
    topology <number>
    kbps <number>
    lcs <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">
  [ disabled_until [<time in seconds> [<by_whom>]]]
  [ 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.20 *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 | kbps
| drop_packet | explicit_ip >
  [ alternate < ga | null | random | ratio | return_to_dns | rr | topology | packet_rate
| leastconn | vs_capacity | static_persist | drop_packet | explicit_ip > ]
  [ fallback < completion_rate | ga | hops | leastconn | packet_rate | qos | random |
ratio | return_to_dns | rr | rtt | topology | vs_capacity | null | static_persist | kbps
| drop_packet | explicit_ip > ]
[ fallback_ip <ip_address> ]
    vs {   //CNAME pools do not require virtual servers
        address <ip_address>:<port number>
      [ ratio <number> ]
      [ limit {
        [ kbytes_per_sec <number> ]
        [ pkts_per_sec <number> ]
        [ current_conns <number> ]
        } ]
      [ depends_on {
          <ip_address>:<port number>
        [ <ip_address>:<port number> ]
        }
      [ disabled_until [<time in seconds> [<by_whom>]]]
    }
  [ vs {
        address <ip_address>:<port number>
    } ]
}
```

Figure A.20 *Syntax for the* ***wideip*** *statement*

Figure A.21 shows an example of a valid **wideip** statement.

```
wideip {
   address           192.168.102.50
   service           "http"
   name              "http.wip.siterequest.com"
   alias             "store.wip.siterequest.com"
   alias             "*.wip.siterequest.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
      vs {
        address 192.168.104.50
        ratio 2
      }
      vs {
        address 192.168.105.50
        ratio 1
      }
   }
}
```

Figure A.21 *Example syntax for the* ***wideip*** *statement*

Definition of wideip sub-statements

The **wideip** sub-statements define pools of virtual servers to be load balanced, and they assign load balancing characteristics, such as the load balancing mode, to each pool. When you have more than one pool configured in a wide IP, the controller first determines the pool that can best respond to a request, and then determines the specific virtual server within the pool that is the best virtual server to send as a response.

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 a unique number, in the IP address format, to identify the wide IP.
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.siterequest.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"**, **"*.siterequest.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.29 *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. If you specify a pool TTL (**pool_ttl**), it overrides the TTL that you specify here. If you do not define either the TTL for the wide IP, or the TTL for the pool, then the controller uses the default TTL (**default_ttl**) that is specified in the globals statement.
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 Controller can send connections to the specified address.
qos_coeff	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 8, *Working with Quality of Service*, in the ***3-DNS Administrator Guide***.

Table A.30 *Load balancing sub-statements*

Parameter	Description
pool_lbmode	Specifies the load balancing mode to use to balance requests over all pools.
manual_resume	Specifies whether disabled virtual servers must be brought back into service manually when they are once again available for load balancing, rather than resuming availability automatically. The default setting is **no**, which indicates that disabled virtual servers resume availability automatically, once the virtual server has successfully responded to service check.

Table A.30 *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 Controller 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**.
file_name	Specifies the name of the object to retrieve.
user	Specifies the user name that you use to log in to the service.
password	Specifies the password that corresponds to the user account.
hashed_password	Specifies the password in encrypted characters.
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.
transfer_timeout	Specifies the maximum amount of time the file information transfer should take.
connection_timeout	Specifies the maximum amount of time to connect to a service.
search_string	Specifies a regular expression that you want the ECV monitor to locate within the scanned file.

Table A.31 *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 Controller 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 system, an EDGE-FX system, or a host.
name	As part of a pool definition, defines the name of the pool. All names must be enclosed in quotation marks, and must be unique within the **wideip** statement.
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.
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 Controller 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 Controller 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 Controller 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 another DNS server.
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 Controller 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 Controller 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 Controller 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 Controller 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.14, on page A-17.

***Table A.32 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 Controller returns the request to DNS. The default is **return_to_dns**. Also see the description of **default_fallback** in Table A.14, on page A-17
fallback_ip	If you specify the **explicit_ip** load balancing mode for the alternate or fallback method, you must also specify the IP address that the controller sends back as the answer to a query. You can use this option to direct traffic to a disaster recovery site.
vs	Specifies the start of a pool virtual server definition. The **vs** sub-statement within the **pool** sub-statement must refer to a virtual server that you defined within a **server** statement. You can use the same virtual server in multiple pools, but not within the same pool.
address	Specifies specifies the IP address and port of the virtual server. The default port is the port of the wide IP, if you do not define a port as part of the **address** parameter.
ratio	Specifies the default weighting to use with respect to all virtual servers in this pool when the Ratio load balancing mode is configured. The default is **1**.
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. See **Limit settings**, on page A-31, for an explanation of the limits you can set.
depends_on	Specifies the IP address and port of other virtual servers that must also be available for load balancing (within this pool) before the 3-DNS Controller uses this virtual server for load balancing.

***Table A.32 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 Controller attempts to load balance requests using the **preferred** mode.
- If the **preferred** mode fails, the 3-DNS Controller tries the **alternate** mode.
- If the **alternate** mode fails, the 3-DNS Controller tries the **fallback** mode.
- If the **fallback** mode fails, the controller tries the next pool. If there are no more pools available, the controller returns the request to DNS. DNS attempts to resolve the request based on the contents of the zone files.

As noted in Table A.33, 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. For information on 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.
drop_packet	The controller drops the request if it cannot find a virtual server to send as a response. We recommend that you use this load balancing mode only in the **fallback** statement.
explicit_ip	Forwards the request to the IP address that you specify in the **fallback_ip** parameter, for example, the IP address for a disaster recovery site. We recommend that you use this load balancing mode only in the **fallback** statement.
global_availability (ga)	Distributes requests to a list of servers, always sending a new request to the first available server in the list.
hops	Sends each new request 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 request to the server that currently hosts the fewest current connections.
null	Bypasses the current load balancing method and forces the 3-DNS Controller 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 request to the server that is managed by a BIG-IP system 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 request: hops, packet rate, completion rate, round trip time, kbps, link capacity, 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 request to a server chosen at random from the pool's virtual servers.
ratio	Distributes new requests across servers in proportion to a user-defined ratio.
return_to_dns	Returns the resolution request to DNS, preventing the 3-DNS Controller from using the next load balancing method or using the next available pool.
rr	Distributes request evenly across all servers, passing each new connection to the next virtual server in line.
rtt	Sends each new request 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 requests 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.

Table A.33 *Load balancing mode sub-statements*

Parameter	Description
static_persist	Distributes requests to a virtual server based on IP address only. The 3-DNS Controller always returns the same virtual server to the same client, if the virtual server is available.
vs_capacity	Distributes requests based on the overall available capacity of the virtual server. Over time all virtual servers in the pool receive requests, but the virtual server with the most capacity receives the highest percentage of requests.
kbps	Distributes requests to the virtual server with the lowest kilobytes per second throughput rate.

Table A.33 *Load balancing mode sub-statements*

Use the following equation to configure the Quality of Service load balancing mode:

```
A (1/packet rate) + B (1/rtt) + C (completion rate) +
D (topology) + E (1/hops) + F (1/kbps) + G (vs_capacity) +
H (link capacity score)
```

Note

For more information about load balancing modes, see Chapter 2, ***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 are 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 in the ***3-DNS Administrator Guide***, Chapter 8, *Configuring a Content Delivery Network*. For more information on topology in general, see Chapter 3, *Topology*, in this guide.

Syntax for the topology statement

Figure A.22 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> ]
 [ <vs_ip_address>/<netmask>  <ldns_ip_address>  <number> ]
}
```

Figure A.22 *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.22.*

Definition of topology sub-statements

The topology sub-statements define the topology records that the 3-DNS Controller 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 Controller 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.
<vs_ip_address>/<netmask>	Specifies a virtual server or group of virtual server, in CIDR format.
<ldns_ip_address>	Specifies a local DNS server.
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 **/var/3dns/include/net.ccdb**. The name must be in quotation marks. Use this for **ldns** in a topology record.

Table A.34 ***Topology*** *sub-statements*

Parameter	Description
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 Controller to evaluate the best resolution option for a DNS request.

***Table A.34 Topology** sub-statements*

Access control lists

You can create access control lists (ACLs) that contain a group of LDNS IP addresses whose paths the 3-DNS Controller 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.

```
region_db ACL {
   region {
      name "probe_acl"
      <ldns cidr>
      <ldns cidr>
   }
   region {
      name "hops_acl"
      region "probe_acl"
       <ldns cidr>
       <ldns cidr>
   }
}
```

***Figure A.23** 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
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 Controller restricts any **big3d** agent from probing the defined group of local DNS servers.
hops_acl	The 3-DNS Controller restricts any **big3d** agent from tracerouting the defined group of local DNS servers

Table A.35 *Access control list sub-statements*

Note

For more information on ACLs, refer to ***Working with access control lists****, on page 5-23.*

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.24 *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.25 *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.26 *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.27 *Syntax for shell style comments*

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 Controller.	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
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 | router | host> show
3dpipe <3dns | bigip | edgefx | router | host> <ip_address> status
3dpipe <3dns | bigip | edgefx | router | host> <ip_address> disable [<time in seconds>
    [<by_whom>]]
3dpipe <3dns | bigip | edgefx | router | host> <ip_address> enable
3dpipe <bigip | edgefx | router | host> <ip_address> virtual show
```

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 current statistics for the 3-DNS Controller.

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 the sync group is enabled or disabled (the status)
- Disable the 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
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
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 wip <wideip_name> pool <pool_name> avail
3dpipe wip <wideip_name> pool <pool_name> virtual show
3dpipe wip <wideip_name> pool <pool_name> virtual <ip_address>:<port> status
3dpipe wip <wideip_name> pool <pool_name> virtual <ip_address>:<port> enable
3dpipe wip <wideip_name> pool <pool_name> virtual <ip_address>:<port> disable [<time in
    seconds> [<by_whom>]]
3dpipe wip <wideip_name> dc <datacenter_name> disable [<time in seconds> [<by_whom>]]
3dpipe wip <wideip_name> dc <datacenter_name> enable
3dpipe wip <wideip_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)
- Enable a virtual server that is disabled in a wide IP pool
- Disable a virtual server in a wide IP pool for a specified number of seconds

- Disable a wide IP, in the context of a data center
- Enable a wide IP that is disabled, in the context of a data center
- View whether a specific a wide IP, in the context of a data center, is enabled or disabled (the status)

C

bigpipe Command Reference

bigpipe commands

This chapter lists the various **bigpipe** commands that are available on the 3-DNS Controller, including syntax requirements and functional descriptions. Note that these commands are BIG-IP system commands that work with the 3-DNS Controller. These commands relate to the BIG-IP-based configuration, which the 3-DNS Controller inherits. 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 Controller 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 BIG-IP system configuration and resets.	C-13

-?

```
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 unit, without copying it to the standby unit. 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 unit.

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 Controller. 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 **bigdb** 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 | show
b global open_failover_ports enable | disable | show
b global open_ftp_ports enable|disable
b global open_rsh_ports enable | disable
b global open_snmp_ports enable | disable | show
b global open_ssh_port enable | disable
b global open_telnet_port enable | disable
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 Controller. 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 Controller in router mode. The default setting is **disabled**.

open_3dns_ports

This variable is required only when running one or more 3-DNS Controllers in the network. It does not apply to running the 3-DNS Controller module on a BIG-IP system.

open_corba_ports

This variable enables and disables the CORBA ports, which allow administrative CORBA connections. The default setting is **disabled**.

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
```

open_ftp_ports

This variable enables or disables ports for FTP access. The default setting is **disable**.

The following command closes FTP ports:

```
b global open_ftp_ports 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 Controller, or if you want a crypto 3-DNS Controller 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_snmp_ports

This variable enables and disables the SNMP ports, which allow administrative SNMP connections. The default setting is **disabled**.

open_ssh_ports

This variable enables or disables ports for SSH access on 3-DNS Controllers 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_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
```

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. To set this logging level, specify a number. The default setting is **0**, representing no logging.

Setting log levels for both UDP and TCP traffic

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 Controller address.

```
b global verbose_log_level 15
```

The following command turns logging off altogether:

```
b global verbose_log_level 0
```

Setting log levels for only TCP traffic

The following command turns on only TCP port denial logging, which logs TCP port denials to the 3-DNS Controller 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 only UDP traffic

The following command turns on only UDP port denial logging, which logs UDP port denials to the 3-DNS Controller 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 show
b interface [<interface_name>] show [verbose]
b interface <inteface_name> media show
b interface <inteface_name> duplex show
b interface <interface_name> media <media_type>
b interface <interface_name> duplex <full | half | auto>
b interface [<interface_name>] stats reset
b interface <interface_name> <enable | disable>
b interface <interface_name> renames <driver_name>
```

Displays the names of installed network interface cards and, for each interface, sets properties such as MAC address, media options, duplex mode, and status, resets interface statistics, enable or disable interfaces, and change driver name mappings.

Options

The **<interface_name>** variable is a name such as **3.1**, where **3** is the physical slot number holding the network interface hardware and **1** is the physical port number on that interface on that hardware.

The **show [verbose]** option displays the current status, settings, and network statistics for the specified interface. The **verbose** argument provides more detailed information. If no interface is specified, this option displays information for all interfaces.

The **media show** option displays information about the media type for the specified interface.

The **duplex show** option displays the duplex mode of the specified interface.

The **media <media_type>** option is a valid media type for the specified interface. Examples include **auto**, **100baseTX**, and **10baseT**. Note that only certain combinations of media type and duplex mode are valid for any particular type of interface.

The **duplex full | half | auto** option sets the duplex mode of the specified interface.

The **stats reset** option resets the statistics for the specified interface.

The **enable | disable** option enables or disables the specified interface.

The **renames <driver_name>** option changes the mapping from the interface's driver name to its physical location name. The **<driver_name> option** is the network interface name in the form of driver and unit number, such as **exp0** and **bs1**. Note that this is the old-style network interface name.

Displaying interface information

To display the status, settings, and statistics for all interfaces on the 3-DNS Controller, use the following command.

```
b interface show [verbose]
```

To display the status, settings, and statistics for a specific interface on the 3-DNS Controller, use the following command-line syntax.

```
b interface <interface_name> show [verbose]
```

Note that if the **verbose** argument is used, the output provides additional information on status. If the **verbose** argument is not used, the output focuses on statistics.

To display the media type for an interface, use the following command-line syntax,

```
b interface <interface_name> media show
```

To display the duplex mode for an interface, use the following command-line syntax.

```
b interface <interface_name> duplex show
```

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 is set to **auto** and the card does not support auto detection, the default type for that interface will be used, for example **1000BaseTX**.

To set the media type, use the following command-line syntax.

```
b interface <interface_name> media <media_type>
```

Setting the duplex mode

Duplex mode may be set to **full**, **half duplex, or auto**. If the media type does not allow **duplex** mode to be set, this is 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 the **bigip.conf** file.

To set the duplex mode, use the following command-line syntax.

```
b interface <interface_name> duplex <full | half | auto>
```

Resetting statistics

You can reset interface statistics for all interfaces or for a specific interface. To reset statistics for all interfaces, use the following command.

```
b interface stats reset
```

To reset statistics for a specific interface, use the following command-line syntax:

```
b interface <interface_name> stats reset
```

Enabling or disabling an interface

Enabling or disabling an interface allows you to control whether the interface receives and sends packets. If an interface begins to behave strangely, you disable and then enable the interface to effectively reset it.

To enable or disable an interface, use the following command-line syntax.

```
b interface <interface_name> enable | disable
```

Changing driver name mapping

You can change the mapping from an interface's driver name to its physical location name, using the following syntax.

```
b interface <interface name> renames <driver name>
```

load

```
b [verify] load [<filename>|-]
b [-log] load [<filename>|-]
```

Resets all of the system management settings, for example, self IP addresses and interfaces, 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/altbigpip.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 base configuration information 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**.

Important

*On a 3-DNS Controller, 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 unit in a redundant system prior to loading a new **/config/bigip.conf** file that contains new timeout values.

For example, you can execute the following commands on a standby 3-DNS unit:

```
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 base configuration and networking 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 the default gateway pool, and floating self IPs for redundant systems. 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 Controllers 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 <ip_addr> vlan <vlan_name | vlangroup_name> [netmask <ip_mask> [broadcast
    <broadcast_addr>]] [unit <id>] [floating <enable | disable>]
b self <ip_addr> vlan <vlan_name | vlangroup_name>
b self <ip_addr> floating enable | disable
b self <ip_addr> delete
b self <ip_addr> show
b self show
```

The **self** command defines a self IP address on a 3-DNS Controller. A self IP address is an IP address mapping to a VLAN or VLAN group and their associated interfaces on a 3-DNS Controller. 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.

Options

The **<ip_addr>** variable specifies an IP address to assign to the 3-DNS Controller.

The **vlan <vlan_name | vlangroup_name>** option specifies the VLAN or VLAN group to which the self IP address is being assigned.

The **netmask <ip mask>** option specifies an IP mask used to set the network of the self IP address.

The **broadcast <broadcast_addr>** option specifies the broadcast address.

The **unit <id>** option specifies an optional unit ID, **1** or **2**. The default value is **1**.

The **floating** option enables or disables a floating self IP address.

Creating self IP addresses

The following are examples of using the **bigpipe self** command to create self IP addresses:

```
b self 10.1.0.1 vlan external netmask 255.255.0.0
b self 10.2.0.1 vlan internal netmask 255.255.0.0
```

For a redundant configuration, the IP addresses that are shared by the two units are configured as floating IP addresses. For example:

```
b self 10.1.1.1 vlan external netmask 255.255.0.0 floating
enable
b self 10.2.1.1 vlan internal netmask 255.255.0.0 floating
enable
```

trunk

```
b trunk <controlling_if> define <if_list>
b trunk [<controlling_if>] show [verbose]
b trunk [<controlling_if>] stats reset
b trunk [<controlling_if>] delete
```

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.

Options

The **<controlling link>** variable specifies the name of the interface chosen to be the controlling link for the trunk. Any attributes of the controlling link at layer 2 and above, such as membership in a VLAN, apply to the trunk.

The **<link>** variable specifies an interface name, for example **3.1**. (For more information on interface naming, refer to the ***3-DNS Administrator Guide***, Chapter 4, *Post-Setup Tasks*.)

The **show** option displays information and statistics for the trunk, on a single line.

The **<verbose>** option, used with the **show** option, displays the information and statistics for the trunk in wordier form.

The **delete** option deletes the specified interface.

unit

```
b unit show
b unit peer show
```

You can use the **bigpipe unit** command to display the unit number assigned to a particular 3-DNS Controller. 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 unit 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 or on a standalone unit, the active unit shows as unit 1 and 2.

verbose

```
b verbose virtual_server_udp_port_denial
b verbose virtual_server_tcp_port_denial
b verbose bigip_udp_port_denial
b verbose bigip_tcp_port_denial
```

Used to modify the verbose log level. This command is an alternative to using the **bigpipe global verbose_log_level** command.

Table C.2 compares use of the **bigpipe verbose** command to use of the **bigpipe global verbose_log_level** command.

b verbose command	b global verbose command
b verbose bigip_udp_port_denial Turns UDP port denial logging **on**. This logs UDP port denials to the 3-DNS system address.	**b global verbose_log_level 1**
b verbose bigip_tcp_port_denial Turns TCP port denial logging **on**. This logs TCP port denials to the 3-DNS system address.	**b global verbose_log_level 2**
b verbose virtual_server_udp_port_denial Turns virtual UDP port denial logging **on**. This logs UDP port denials to the virtual server address.	**b global verbose_log_level 4**
b verbose virtual_server_tcp_port_denial Turns virtual TCP port denial logging **on**. This logs TCP port denials to the virtual server address.	**b global verbose_log_level 8**
b verbose bigip_udp_port_denial **b verbose bigip_tcp_port_denial** **b verbose bigip_udp_port_denial** **b verbose bigip_tcp_port_denial** Turns UDP and TCP port denial on for both virtual server and 3-DNS system addresses.	**b global verbose_log_level 15**

***Table C.2 bigpipe verbose** and **global verbose** command equivalencies*

verify

```
b [log] verify <command...]
b verify load [<filename>|-]
```

The **verify** command 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/altbigpip.conf**, use the following command:

```
b verify load /config/altbigip.conf
```

version

```
b version
```

Displays the version of the 3-DNS Controller operating system and the features that are enabled.

For example, for a 3-DNS Controller, the **bigpipe version** command displays the output shown in Figure C.1.

```
Product Code:
3-DNS

Enabled Features:
BIG_IP Link Control                    3-DNS (R)
Pools                                  Failover
Health Check                           Filter
3-DNS Engine                           3-DNS Multiple Pools
Statistics Journaling                  Network Proximity Table
IP Classifier                          Internet Weather Map
...
```

Figure C.1 *The **version** output display*

vlan

```
b vlan <vlan_name>
b vlan <vlan_name> rename <new_vlan_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 | 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 <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 Controller 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.

Options

The **<vlan name>** variable specifies a VLAN name, 1-15 characters in length.

The **tag <tag number>** option specifies a valid VLAN tag number, in the range **0**-**4095**. Note that if **0** is specified as the tag number, the **vlan** command creates an empty VLAN.

The **interfaces add [tagged]** option specifies that the interfaces specified with the **<if_list>** argument are to be added to the specified VLAN, as either tagged or untagged interfaces.

The **interfaces delete** option deletes all interfaces for the specified VLAN.

The **<if_list>** variable specifies a list of interfaces to be added to a VLAN.

The **interfaces show all** option shows all interfaces for the specified VLAN.

The **failsafe** option allows you to arm, disarm, or show the failsafe mechanism for redundant systems.

The **timeout <timeout>** option specifies a timeout value for the failsafe mechanism.

The **rename <new_vlan_name>** option specifies the name to which you want to rename the specified VLAN.

The **<if_name>** variable specifies an interface name.

The **mac_masq <MAC address>** option specifies a MAC address, such as **0:a0:be:ef:1f:3a,** that will be shared by both units in a redundant system.

The **port_lockdown** option enables or disables connections through the specified VLAN.

vlangroup

```
b vlangroup [<vlangroup name>] <show | list | delete>
b vlangroup <vlan name> tag <number>
b vlangroup [<vlangroup name>] tag [show]
b vlangroup [<vlangroup name>] interfaces [show]
b vlangroup <vlan name> vlans add <vlan if name list>
b vlangroup <vlangroup name> vlans delete <vlan if name list | all>
b vlangroup [<vlangroup name>] vlans [show]
b vlangroup <vlangroup name> port_lockdown <enable | disable | show>
b vlangroup <vlangroup name> proxy_forward <enable | disable>
b vlangroup [<vlangroup name>] proxy_forward [show]
b vlangroup <vlangroup name> failsafe <arm | disarm | show>
b vlangroup <vlangroup name> timeout <number>
b vlangroup <vlangroup name> mac_masq <MAC addr | show>
b vlangroup <vlangroup name> fdb <add | delete> <MAC addr> interface <if name>
b vlangroup [<vlangroup name>] fdb <show [static | dynamic]>
b vlangroup <vlan name> rename <vlan name>
```

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.

Note that if a VLAN belongs to multiple VLAN groups, you can only delete the VLAN from one VLAN group at a time.

Options

The arguments available with the **bigpipe vlangroup** command are the same as those for the **bigpipe vlan** command. For a description of these options, see *Options*, on page C-24.

D

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 D.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 D.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 D.1 shows an example of an **A** record.

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

Figure D.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 D.2 shows an example of a **CNAME** resource record.

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

***Figure D.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 D.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 D.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 D.4 shows an example of an **NS** resource record.

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

***Figure D.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 D.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 D.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.

Figure D.6 shows an example of an **SOA** record.

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

Figure D.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,`** **johndoe@berkeley.edu** becomes **johndoe.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 D.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 D.2 *Less common resource records*

E

The bigdb Database and bigdb Keys

- Working with the bigdb database
- Supported bigdb keys on a 3-DNS Controller

Working with the bigdb database

The bigdb™ database holds certain configuration information for the 3-DNS Controller. You use the **bigpipe db** command to load configuration information into the bigdb database. The **default.txt** file contains the default information for the bigdb database.

Using the bigpipe db command

You can view and set the bigdb keys using following options of the **bigpipe db** command. Note that these keys are set automatically when you use the Configuration utility to configure the 3-DNS Controller.

To display the current setting of a bigdb configuration key

To display the value of a bigdb key, use the following syntax:

```
b db get <key | regular_exp>
```

For example, the following command displays the value of the **Local.Bigip.Failover.ForceActive** key, which causes the unit to always be the active unit in a redundant system. By default, this key is set to **0** (zero), which is off.

```
b db get Local.Bigip.Failover.ForceActive
```

You can use the following command to display the value of all local keys:

```
b db get Local.*
```

To set a bigdb configuration key

To create (set) a bigdb key, use the following syntax:

```
b db set <key>
```

To create a bigdb key and also assign a value to it, use the following syntax:

```
b db set <key> = <value>
```

For example, the following command causes the unit to always try to be the active unit in a redundant system:

```
b db set Local.Bigip.Failover.ForceActive = 1
```

To disable a bigdb key

To disable a bigdb key, use the following syntax.

```
b db unset <key | regular_exp>
```

For example, the following command disables the **Local.Bigip.Failover.UnitId** key:

```
b db unset Local.Bigip.Failover.UnitId
```

The following command disables all local keys:

```
b db unset set Local.*
```

Working with the default.txt file

The **default.txt** file documents the keys that are valid in the bigdb database. This file is located in the **/config/default.txt** directory. It contains all the possible database keys, comments that document these keys, and the default values used by programs that run on the 3-DNS Controller.

◆ Note

*The values in the **default.txt** file are default values; several of the keys listed are not present in the bigdb database.*

The **default.txt** file is intended to serve as documentation only. In order for the system to work, some of the keys, such as those that represent IP addresses and port numbers, need to be set to values other than the default values. Additionally, some of the key names listed are wildcard keys. These keys are not valid key names.

For a list of the keys in the bigdb database that are applicable to a 3-DNS Controller, see the following section.

Supported bigdb keys on a 3-DNS Controller

The bigdb database contains failover and system keys for the 3-DNS Controller. Configuration options in the bigdb database that the 3-DNS Controller supports include:

- Fail-over
- CORBA

The bigdb keys for each of these features are described in the following series of tables. You can view and set the bigdb keys using the **bigpipe db** command, as described in *Using the bigpipe db command*, on page E-1.

Fail-over and cluster keys

The fail-over and cluster keys control fail-over from the active to the standby unit in a 3-DNS redundant system. If you change one of these values, you must update the 3-DNS configuration using the **bigpipe failover init** command (which is the same command as **bigstart reinit sod**). This command forces the system to reread the bigdb database and use the new values. To run this command, type the following:

```
b failover init
```

Table E.1, on page E-4, lists the fail-over and cluster keys.

Fail-Over Key Name and Default Value	Description
Common.Bigip.Failover.AwaitPeerAliveDelay = 2	Delay in seconds before testing whether peer is active. The default value is **2**.
Common.Bigip.Failover.AwaitPeerDeadDelay = 1	Delay in seconds before testing whether the peer has failed. The default value is **1**.
Common.Bigip.Failover.NoSyncTime	By default, one unit in a redundant system synchronizes its time with the other unit. Set this key to **1** to turn off the time synchronization feature.
Common.Bigip.Failover.DbgFile	File into which **sod** logs the fail-over debug information.
Common.Bigip.Failover.PrintPeerState = 0	The default value for this key is **0**. The fail-over utility writes the state of its connection to its peer. This information is written to the fail-over utility's debug log file.
Common.Bigip.Failover.UseTty00 = 0	The fail-over utility uses **/dev/tty00** for hard wired fail-over.
Common.Bigip.Failover.UseTty01 = 1	The fail-over utility uses **/dev/tty01** for hard wired fail-over.
Local.Bigip.Failover.ForceActive = 0	This unit always attempts to become the active unit in a redundant system.
Local.Bigip.Failover.ForceStandby = 0	This unit goes to standby whenever it senses that its peer is alive.
Common.Sys.Failover.Network = 0	Use the network as a backup to, or instead of, the serial line for fail-over if this value is **1**. By default, this feature is **0** (off).
Common.Sys.Failover.NetTimeoutSec = 3	When using network failover, after sending a ping to this unit's peer, wait this many seconds for a response to the ping before timing out and sending another ping.
Common.Sys.Failover.NetRetryLimit = 3	When using network failover, if the ping times out, then resend the ping this many times before initiating a reboot for the peer unit.
Common.Bigip.Failover.On3DNSFail = true	This variable, if set, will cause a fail-over if the **3dnsd** daemon dies.

Table E.1 *The bigdb keys that store fail-over information*

CORBA keys

Table E.2 lists the CORBA keys contained in the bigdb database.

Key Names	Description
Common.Bigip.CORBA.IIOPPort ="683"	Default CORBA IIOP port used for the iControl™ API.
Common.Bigip.CORBA.SSLPort ="684"	Default CORBA IIOP SSL port used for the iControl™ API.
Common.Bigip.CORBA.AddrResolveNumeric="true"	When set to **true**, the CORBA portal resolves client addresses numerically.

Table E.2 *bigdb keys for CORBA settings*

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 Controller.

3-DNS Maintenance menu

The 3-DNS Maintenance menu is a command line utility that you use to configure the 3-DNS Controller.

3-DNS web server

The 3-DNS web server is a standard web server that hosts the Configuration utility on the 3-DNS Controller.

A record

The **A** record is the ADDRESS resource record that a 3-DNS Controller 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 Controller 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 Controller 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 Controller that make CDN switching possible are geographic redirection and CNAME pools.

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 Controller.

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 Controllers, BIG-IP systems, EDGE-FX Caches, or host machines.

data center server

A data center server is any server recognized in the 3-DNS Controller configuration. A data center server can be any of the following: a 3-DNS Controller, a BIG-IP system, an EDGE-FX Cache, 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 Controller, 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 Controller uses to load balance a resolution request. The 3-DNS Controller 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 Controller 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 Controllers.

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 Controllers, BIG-IP systems, and EDGE-FX Caches. 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 Controller uses three load balancing modes. The preferred method specifies the first load balancing mode that the 3-DNS Controller 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 Controller 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 Controller, 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, 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 Controllers 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 Controllers 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 Controller, 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 Controller 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 system, an EDGE-FX Cache, or a host. The 3-DNS Controller 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 Controller 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 Controller uses to load balance a resolution request. See also *alternate method, fallback method*.

principal controller

The principal controller is the 3-DNS Controller that initiates metrics collection by the **big3d** agents, auto-discovers objects in the network, and is the preferred system on which to make configuration changes for a sync group. All sync group members (both the principal and receivers) receive broadcasts of metrics data from the **big3d** agents throughout the network. All sync group members also receive broadcasts of updated configuration settings from the sync group member that has the latest configuration changes. See also *receiver controller, sync group*.

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 Controller 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 Controller. Prober factories can run on BIG-IP systems and EDGE-FX Caches.

production rule

A production rule, on the 3-DNS Controller, 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, link capacity, 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 controller

A receiver controller is a sync group member that receives auto-discovery updates from the principal 3-DNS Controller. All sync group members (both the principal and receivers) receive broadcasts of metrics data from the **big3d** agents throughout the network. All sync group members also receive broadcasts of updated configuration settings from the sync group member that has the latest configuration changes. See also *principal controller, sync group*.

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

A 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 Controllers that synchronize system configurations and zone files (if applicable). Sync groups have one principal controller, and may contain one or more receiver controllers. All sync group members (both the principal and receivers) receive broadcasts of metrics data from the **big3d** agents throughout the network. All sync group members also receive broadcasts of updated configuration settings from the 3-DNS Controller that has the latest configuration changes. See also *principal controller, receiver controller*.

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 Controller 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 Controller 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 Controller and does not yet have metrics information, the 3-DNS Controller marks its status as **unknown**. The 3-DNS Controller can use unknown servers for load balancing, but if the load balancing mode is dynamic, the 3-DNS Controller 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 system, EDGE-FX Cache, or host server.

watchdog timer card

The watchdog timer card is a hardware device that monitors the 3-DNS Controller 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 Controller 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

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z