

# 3-DNS<sup>®</sup> Distributed Traffic Controller Reference Guide

version 3.0







---

# Service and Support Information

## Product Version

This manual applies to version 3.0 of the 3-DNS® Controller.

## Obtaining Technical Support

<b>Web</b>	tech.f5.com
<b>Phone</b>	(206) 272-0888
<b>Fax</b>	(206) 272-0802
<b>Email (support issues)</b>	support@f5.com
<b>Email (suggestions)</b>	feedback@f5.com

## Contacting F5 Networks

<b>Web</b>	www.f5.com
<b>Toll-free phone</b>	(888) 88BIG-IP
<b>Corporate phone</b>	(206) 272-5555
<b>Fax</b>	(206) 272-5556
<b>Email</b>	sales@f5.com
<b>Mailing Address</b>	401 Elliott Avenue West Seattle, Washington 98119

---

## 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 herein. F5 reserves the right to change specifications or documentation at any time without notice.

Copyright ©1999-2001, F5 Networks, Inc. All rights reserved.

### Trademarks

F5, BIG-IP, and 3-DNS are registered trademarks of F5 Networks, Inc. SEE-IT, GLOBAL-SITE, EDGE-FX, and FireGuard are trademarks of F5 Networks, Inc. In Japan, F5 is trademark number 4386949 and SEE-IT is trademark number 4394516. All other product and company names are registered trademarks or trademarks of their respective holders.

### Export Regulation Notice

The 3-DNS® Controller may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this 3-DNS® Controller 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 State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.

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

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

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

This product includes software developed by the Apache Group for use in the Apache HTTP server project

---

(<http://www.apache.org/>).

This product includes software developed by Darren Reed. (© 1993-1998 by Darren Reed).

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

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

This product includes software developed by Eric Young.

Portions of the material included in Appendix C came from the Internet Software Consortium.  
<http://www.isc.org/>.

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 ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)), (© 1995-1998).

This product includes open SSH software developed by Tatu Ylonen <[ylo@cs.hut.fi](mailto: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](http://www.mindbright.se), [info@mindbright.se](mailto:info@mindbright.se) (© 1998-1999).

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

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



---

---

# Table of Contents

---

---





---

<b>1</b>		
	<b>Introduction</b>	
	Getting started .....	1-1
	Using the Administrator Kit .....	1-1
	Stylistic conventions .....	1-2
	Finding help and technical support resources .....	1-3
<b>2</b>		
	<b>Access Control Lists</b>	
	Access control lists .....	2-1
<b>3</b>		
	<b>big3d Agent</b>	
	big3d agent .....	3-1
	Setting up data collection with the big3d agent .....	3-1
	Installing the big3d agent .....	3-3
	Understanding factories run by big3d agents .....	3-3
	Setting up communication between 3-DNS Controllers and BIG-IP Controllers .....	3-6
<b>4</b>		
	<b>Extended Content Verification (ECV)</b>	
	Extended Content Verification (ECV) .....	4-1
<b>5</b>		
	<b>Load Balancing</b>	
	Load balancing modes .....	5-1
	Understanding load balancing .....	5-1
	Using static load balancing modes .....	5-3
	Using dynamic load balancing modes .....	5-7
	Configuring load balancing .....	5-13
	Understanding wide IPs .....	5-13
	Understanding pools .....	5-14
	Adding a wide IP .....	5-14
	Changing global variables that affect load balancing .....	5-20
	Setting global alternate and fallback modes .....	5-21
	Understanding TTL and timer values .....	5-22

## 6

### Network Map

Network Map .....	6-1
Working with the Network Map .....	6-2
Using the information table on the Network Map .....	6-3

## 7

### Production Rules

Controlling network traffic patterns with production rules .....	7-1
Setting up production rules in the Configuration utility .....	7-1
Viewing, adding, and deleting production rules .....	7-2
Choosing the rule type .....	7-2
Defining time-based triggers .....	7-3
Defining event-based triggers .....	7-5
Choosing the action .....	7-7
Working with the production rules scripting language .....	7-8
Inserting production rules in the wideip.conf file .....	7-8
Executing and managing production rules .....	7-8
The if statement .....	7-9
The when statement .....	7-11
The every statement .....	7-12
Production rule actions .....	7-13
Production rule examples .....	7-14

## 8

### Resource Records

Resource records .....	8-1
Types of resource records .....	8-2
Additional resource record types .....	8-5

## 9

### Scripts

Scripts .....	9-1
3dns_admin_start .....	9-1
3dns_auth .....	9-1
3dns_dump .....	9-1
3dns_sync_metrics .....	9-2
3dns_web_config .....	9-2
3dns_web_passwd .....	9-2
3dnsmaint .....	9-2

3dprint .....	9-3
3ndc .....	9-4
big3d_check .....	9-5
big3d_install .....	9-5
big3d_restart .....	9-6
big3d_version .....	9-6
edit_lock .....	9-6
edit_wideip .....	9-6
install_key and F5makekey .....	9-7
syncd_checkpoint .....	9-7
syncd_rollback .....	9-8
syncd_start .....	9-9
syncd_stop .....	9-10

## 10

### SNMP

Working with SNMP on the 3-DNS Controller .....	10-1
Configuring SNMP on the 3-DNS Controller .....	10-1
Downloading the MIBs .....	10-2
Understanding configuration file requirements .....	10-3
/etc/hosts.deny .....	10-3
/etc/hosts.allow .....	10-3
/etc/snmpd.conf .....	10-4
/etc/snmptrap.conf .....	10-6
/etc/syslog.conf .....	10-7
Configuring options for the checktrap script .....	10-7
Options for checktrap .....	10-7
Configuring the 3-DNS SNMP agent using the Configuration utility .....	10-8
To set SNMP properties using the Configuration utility .....	10-9
Configuring host SNMP settings on the 3-DNS Controller .....	10-9
Configuring SNMP agents on hosts .....	10-11

## 11

### Topology

Topology .....	11-1
Setting up topology records .....	11-1

Table of Contents

---

12

Utilities

Utilities ..... 12-1  
3-DNS Controller utilities documentation ..... 12-1

Index



---

---

# Introduction

---

---

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





---

## Getting started

The *3-DNS Controller 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 Reference Guide for help in configuring a specific feature of the 3-DNS Controller. For load balancing and networking solutions, see the Administrator Guide.

## Using the Administrator Kit

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

- ◆ **Installation Guide**

The *3-DNS Controller Installation Guide* walks you through the basic steps needed to get the hardware plugged in and the system connected to the network. Most users turn to this guide only the first time that they set up a 3-DNS Controller. The Installation Guide also covers general network administration issues, such as setting up common network administration tools including Sendmail.

- ◆ **Administrator Guide**

The *3-DNS Controller 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.

- ◆ **Reference Guide**

The *3-DNS Controller Reference Guide* provides basic descriptions of individual 3-DNS Controller objects, such as

wide IPs, pools, and virtual servers. It also provides syntax information for **3dnsmaint** commands, configuration utilities, configuration files, and system utilities, scripts and SNMP.

### Stylistic conventions

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

#### **WARNING**

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

### Identifying new terms

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

### Identifying references to objects, names, and commands

We make a variety of items bold to help you easily pick them out of a block of text. These items include web addresses, IP addresses, utility names, and portions of commands, such as variables and keywords. For example, the **nslookup** command requires that you include at least one **<ip\_address>** variable.

### Identifying references

We use italic text to denote a reference to another document or another section in the current document. In references where we provide the name of a book as well as a specific chapter or section in the book, we show the book name in bold, italic text, and the chapter/section name in italic text to help quickly differentiate the

two. For example, you can find information about **3dnsmaint** commands in the section *3dnsmaint*, on page 9-2.

### 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:

```
ndc 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 <b>&lt;your name&gt;</b> , 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 Controller 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 ndc**), 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 and GateD.

◆ **Technical support through the World Wide Web**

The F5 Networks Technical Support web site, <http://tech.F5.com>, contains the Ask F5 knowledge base and provides the latest technical notes and updates for the *Administrator Kit* guides (in PDF and HTML formats). To access this site you must first email [askf5@f5.com](mailto:askf5@f5.com) to obtain a customer ID and a password.

# 2

---

---

## Access Control Lists

---

---





## Access control lists

With access control lists (ACLs), you can block probing for members of the ACL when you use dynamic RTT probing on your 3-DNS Controller. Table 2.1 lists the ACL types and describes their functions.

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

*Table 2.1 Access control list types and descriptions*

### To define ACLs using the Configuration utility

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

## To define ACLs using the command line

### ◆ Tip

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

1. If one does not already exist, create a file called **region.ACL** in the **/var/3dns/include** directory. You must add the include file at the beginning of the *wideip.conf* file.
2. Add the file to **/etc/wideip.conf** by typing, at the command line:  
`include "region.ACL"`
3. The ACLs you can create are **probe\_acl**, **hops\_acl**, and **discovery\_acl**. Figure 2.1 is an example the syntax for a **region.ACL** file with definitions for the three ACL types.

```
actions {
    NO_RELAY
    delete rdb ACL region "probe_acl"
    delete rdb ACL region "hops_acl"
    delete rdb ACL region "discovery_acl"
}
region_db ACL {
    region {
        name "probe_acl"
        region "probe_acl"
        192.168.4.0/24
    }
    region {
        name "hops_acl"
        192.168.2.0/16
    }
    region {
        name "discovery_acl"
        192.168.11.11/32
        192.168.4.0/24
    }
}
```

*Figure 2.1* Sample *region.ACL* file

# 3

---

---

## big3d Agent

---

---





## 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 Controllers, and EDGE-FX Caches; the default setting is to run a **big3d** agent on all of these controllers in the network, but you can turn off the **big3d** agent on any controller at any time.

### Setting up data collection with the big3d agent

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

- ◆ **Installing big3d agents on BIG-IP Controllers and EDGE-FX Caches**

Each new version of the 3-DNS Controller software includes the latest version of the **big3d** agent. You need to distribute that copy of the **big3d** agent to the BIG-IP Controllers and EDGE-FX Caches in the network. See the release notes provided with the 3-DNS Controller software for information about which BIG-IP Controller and EDGE-FX Cache versions the current **big3d** agent supports. For details on installing the **big3d** agent, see *Installing the big3d agent*, on page 3-3.
- ◆ **Specifying which factories a specific big3d agent manages**

When you define BIG-IP Controllers, EDGE-FX Caches, and 3-DNS Controller servers, you can change the default **big3d** agent settings on a specific controller. You can change the number of factories the **big3d** agent runs and turn specific factories on and off.
- ◆ **Setting up communications between big3d agents and controllers**

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 BIG-IP Controllers*, on page 3-6.

## Path data and server performance

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

- ◆ **Virtual server availability**

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

- ◆ **Network path round trip time**

The **big3d** agent calculates the round trip time for the network path between the data center and the client's LDNS server that is making the resolution request. Round trip time is used in determining the best virtual server when using the Round Trip Times or the Quality of Service modes.

- ◆ **Network path packet loss**

The **big3d** agent calculates the packet completion percentage for the network path between the data center and the client's LDNS server that is making the resolution request. Packet completion is used in determining the best virtual server when using the Completion Rate or the Quality of Service modes.

- ◆ **Hops along the network path**

The **big3d** agent calculates the number of intermediate systems transitions (hops) between the data center and the client's LDNS server. Hops are used in determining the best virtual server when using the Hops or the Quality of Service load balancing modes.

- ◆ **Server performance**

The **big3d** agent calculates the packet rate of the BIG-IP Controller or SNMP-enabled hosts. Packet rate is used in determining the best virtual server when using the Packet Rate or the Quality of Service load balancing modes.

- ◆ **Virtual server performance**

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

## Installing the big3d agent

You can easily install the **big3d** agent on the BIG-IP Controllers and EDGE-FX Caches in your network by using the **3dnsmaint** command line utility.

### To install the big3d agent using the command line

1. Log on to the 3-DNS Controller using either a remote shell, a serial terminal, or the keyboard and monitor attached directly to the controller.
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 which collects different types of data. The **big3d** agent currently supports five factory types:

### ◆ Probing factory

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

### ◆ Hops factory

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

### ◆ SNMP factory

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

- ◆ **Discovery factory**

A discovery factory acts as a backup to a probing factory. The **big3d** agent runs a discovery factory only when a probing factory fails to get a response from a specific LDNS. The **big3d** agent uses the discovery factory to look for an alternate port on an LDNS that can respond to the queries issued by a probing factory. If the discovery factory finds an open port, it returns the port number to the 3-DNS Controller, which stores the number to use for future path probe attempts.

- ◆ **Permanent factories**

Two permanent factories collect performance information. One factory collects information from the BIG-IP Controller when it exists, the other collects the number of packets being processed per second. These factories are not configurable.

The standard configuration specifies that each BIG-IP Controller, EDGE-FX Cache, and 3-DNS Controller in the network run a **big3d** agent using five prober factories, one SNMP factory, one discovery factory, no hops factories, and the two permanent factories. In the BIG-IP Controller or 3-DNS Controller server definition, you can change the number of factories that the **big3d** agent runs. For example, the default number of hops factories is set to **0**; if you want to run a hops factory, you change the setting to **1** or more.

### Understanding the data collection and broadcasting sequence

The **big3d** agents collect and broadcast information on demand. The principal 3-DNS Controller in the 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 the factories, and then broadcast that data to all 3-DNS Controllers running in the network, including the principal controller that issued the request.

## Important notes about 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 controller. Table 3.1 shows the six states that can be assigned to an LDNS. Note that you can view the state of LDNS servers in the Local DNS Statistics page in the Configuration utility.

State	Description
<b>Needs Probe</b>	The <b>big3d</b> agent has never collected data for the LDNS, or the data has expired.
<b>Idle</b>	The <b>big3d</b> agent successfully collected data for the LDNS, and is waiting for the next collection request.
<b>In Probe</b>	The <b>big3d</b> agent is currently collecting data for the LDNS.
<b>Needs Discovery</b>	The <b>big3d</b> agent failed to collect data for the LDNS using its standard protocols and ports, and now needs to run the LDNS through a discovery factory.
<b>In Discovery</b>	The <b>big3d</b> agent is currently running the LDNS through a discovery factory to look for an alternate available port.
<b>Suspended</b>	The <b>big3d</b> agent failed to discover a port open for data collection, and the LDNS is no longer eligible for probing.

*Table 3.1 Probe and discovery states for individual client LDNS servers*

## Evaluating big3d agent configuration trade-offs

You must run a **big3d** agent on each BIG-IP Controller, 3-DNS Controller, and EDGE-FX Cache. If you are using advanced load balancing modes, you must have a **big3d** agent running on at least one controller 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 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 paths that the **big3d** agent has 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 BIG-IP Controllers

In order to copy **big3d** agents from the 3-DNS Controllers to the BIG-IP Controllers and the EDGE-FX Caches, the 3-DNS Controllers need to communicate with BIG-IP Controllers. If you use exclusively crypto 3-DNS Controllers and crypto BIG-IP Controllers, or exclusively non-crypto 3-DNS Controllers and non-crypto BIG-IP Controllers, the communication tools set up by the First-Time Boot utility are all you need. Crypto controllers all use **ssh** and **scp**, and non-crypto controllers all use **rsh** and **rcp**.

However, if you work in a mixed environment where some controllers are crypto, and other controllers are non-crypto, you need to enable the **rsh** and **rcp** tools on the crypto controllers. These tools come pre-installed on all crypto 3-DNS and BIG-IP Controllers, but you must explicitly enable them.

## To enable the rlogin tools on a BIG-IP Controller

From the command line, run the **rsetup** script.

### ◆ Note

You can disable **rsh** and **rcp** access at any time either by running the **config\_rshd** script, or by changing the **bigip.open\_rsh\_ports** system control variable to **0** in **/etc/rc.sysctl**.

Table 3.2 shows the ports and protocols that 3-DNS Controllers use to communicate with BIG-IP Controllers and EDGE-FX Caches.

From	To	Protocol	From Port	To Port	Purpose
Crypto 3-DNS Controller	Crypto BIG-IP Controller	TCP	<1023	22	SSH/SCP
Non-crypto 3-DNS Controller	Non-crypto BIG-IP Controller	TCP	>1024	514	RSH/RCP
Crypto 3-DNS Controller	Non-crypto BIG-IP Controller	TCP	>1024	514	RSH/RCP
Non-crypto BIG-IP Controller	Crypto 3-DNS Controller	N/A	N/A	N/A	N/A

**Table 3.2** Communications between 3-DNS Controllers and BIG-IP Controllers

Note that if you run **big3d** agents in a mixed crypto/non-crypto environment, the crypto controllers automatically turn off Blowfish encryption when communicating with non-crypto **big3d** agents. When communicating with crypto **big3d** agents, however, crypto 3-DNS Controllers always use Blowfish encryption by default, although you can manually disable it if you prefer.

## Setting up iQuery communications for the big3d agent

The iQuery protocol can use one of two ports to communicate between the **big3d** agents and the 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.

Figure 3.1 shows the ports necessary for iQuery communication between 3-DNS Controllers and **big3d** agents that run on 3-DNS Controllers, BIG-IP Controllers, or EDGE-FX Caches.

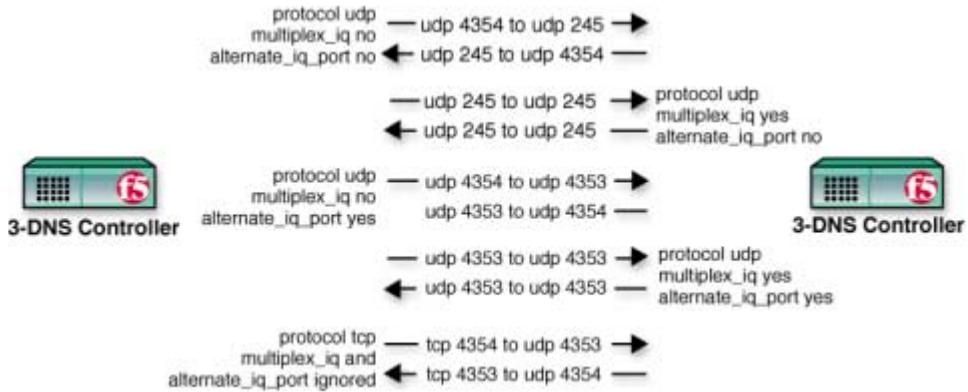


Figure 3.1 iQuery communication

Table 3.3 shows the port numbers and corresponding protocols used for iQuery traffic.

From	To	Protocol	From Port	To Port	Purpose
3-DNS Controller	big3d agent	UDP	245, 4354	245	Old standard iQuery port for outbound traffic
3-DNS Controller	big3d agent	UDP TCP	4353 or 4354	4354 or 4353	Alternate iQuery port for outbound traffic (open this port only when the <b>use_alternate_iq</b> global variable is set to <b>yes</b> )
<b>big3d</b> agent	3-DNS Controller	UDP TCP	245 or 4353	4354	Ephemeral port used for inbound iQuery traffic (when <b>multiplex_iq</b> is set to <b>no</b> )

Table 3.3 Communications between 3-DNS Controllers, **big3d** agents, and host servers

From	To	Protocol	From Port	To Port	Purpose
big3d agent	3-DNS Controller	UDP	245	245	Single port used for multiplexed inbound iQuery traffic (open this port only when the <b>multiplex_iq</b> global variable is set to <b>yes</b> )
big3d agent	3-DNS Controller	UDP	4353	4353	Single port used for multiplexed inbound iQuery traffic (open this port only when both the <b>use_alternate_iq</b> and the <b>multiplex_iq</b> global variables are set to <b>yes</b> )
		TCP	4354 4353	4353 4354	
big3d agent	host SNMP agent	UDP	>1024	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 via SNMP

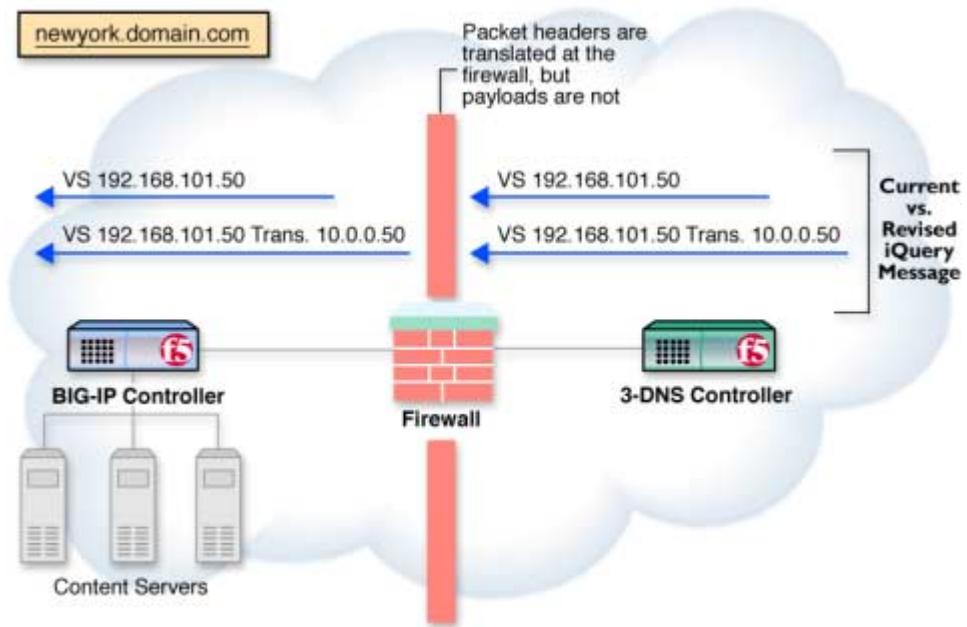
**Table 3.3** Communications between 3-DNS Controllers, **big3d** agents, and host servers

### Allowing iQuery communications to pass through firewalls

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

The iQuery translation option resolves this issue. With iQuery translation turned on, 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 3.2, a firewall separates the path between a BIG-IP Controller 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 Controller in their original states.



*Figure 3.2* Translating packet address through the firewall

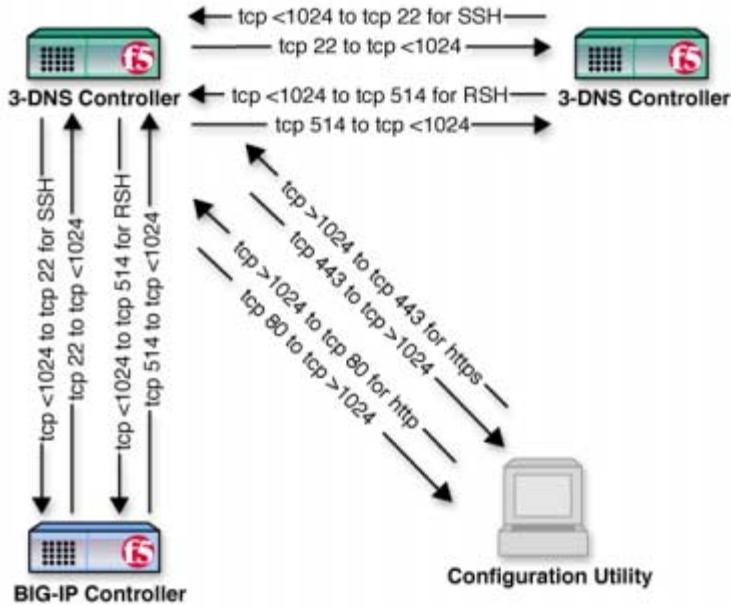
## Communications between 3-DNS Controllers and other machines in the network

The following tables show the other ports and protocols that 3-DNS Controllers use for communication. Table 3.4 shows the ports that 3-DNS Controllers use for remote administrative connections to the 3-DNS web server.

From	To	Protocol	Port	Purpose
Remote Workstation	Crypto 3-DNS Controller	TCP	443	Connection to secure web server
Remote Workstation	Non-crypto 3-DNS Controller	TCP	80	Connection to standard web server

**Table 3.4** *Communications between 3-DNS Controllers and remote workstations*

Figure 3.3 shows the ports necessary for administrative communication between individual 3-DNS Controllers, and also between 3-DNS Controllers and administrative workstations.



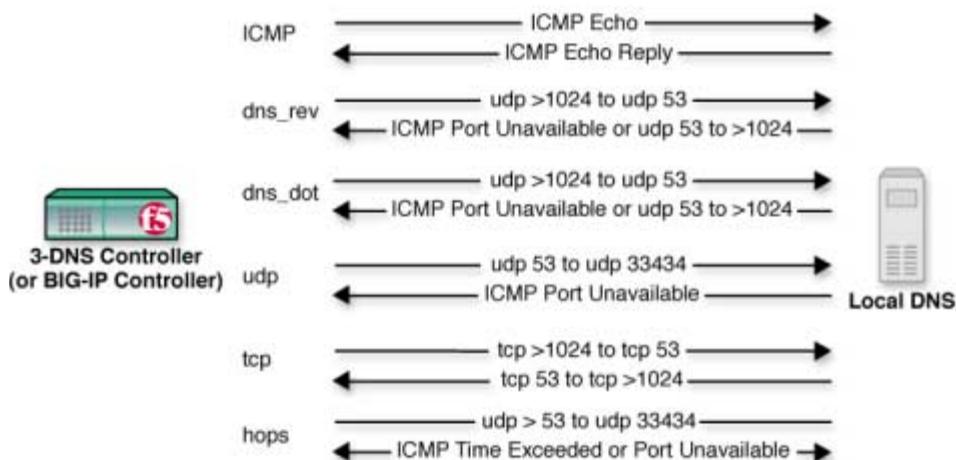
**Figure 3.3** Administrative communications ports

Table 3.5 shows the ports on which the 3-DNS Controller receives and responds to DNS resolution requests issued by LDNS servers.

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

**Table 3.5** DNS communications on the 3-DNS Controller

Figure 3.4 shows the ports necessary for path probing between 3-DNS Controllers and local DNS servers.



**Figure 3.4** Path probe communications

Table 3.6 shows the ports that the **big3d** agent uses when collecting path data for LDNS servers.

From	To	Protocol	From Port	To Port	Purpose
big3d agent	LDNS	ICMP	N/A	N/A	Probing using ICMP pings
big3d agent	LDNS	TCP	2000-12000	53	Probing using TCP (CISCO routers should "allow establish")
LDNS	<b>big3d</b> agent	TCP	53	2000-12000	Probing using TCP (CISCO routers should "allow establish")
<b>big3d</b> agent	LDNS	UDP	2000-12000	33434	UDP probing and traceroute

**Table 3.6** Communication between **big3d** agents and LDNS servers

From	To	Protocol	From Port	To Port	Purpose
LDNS	<b>big3d</b> agent	ICMP	N/A	N/A	Replies from ICMP, UDP pings, or traceroute probes
<b>big3d</b> agent	LDNS	<b>dns_rev</b> <b>dns_dot</b>	>1024	53	DNS version or dot queries
LDNS	<b>big3d</b> agent	<b>dns_rev</b> <b>dns_dot</b>	53	>1024	DNS version or dot response

**Table 3.6** Communication between **big3d** agents and LDNS servers

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

# 4

---

---

## Extended Content Verification (ECV)

---

---





## Extended Content Verification (ECV)

If you set up an ECV service monitor, 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 via the HTTP, HTTPS, or FTP network services.

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.

The way ECV works is, if the file responds appropriately to the ECV query, the server where the file resides is marked as **up** and the client will be sent to that server. If the file does not respond as expected to the ECV query, the server where the file resides is marked as **down**, and the client will not be sent to that server.

### To define ECV service monitors using the Configuration utility

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

### To define ECV service monitors from the command line

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

3. Use the syntax shown in Figure 4.1 to define an ECV service monitor.

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

◆ **Note**

*The https protocol is available only on crypto 3-DNS Controllers.*

```
ecv {
    protocol      <http | https | ftp>
    filename      "<path and file name>"
    scan_level    <all | first>
    user          "<user name for servers that require authentication>"
    hashed_password "<hashed version of server password>"
}
```

**Figure 4.1** Syntax for defining ECV service monitors

Figure 4.2 shows a sample ECV statement that defines an ECV service monitor.

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

**Figure 4.2** Sample ECV service monitor definition

# 5

---

---

## Load Balancing

---

---

- Load balancing modes





## Load balancing modes

The 3-DNS Controllers use the load balancing modes when resolving DNS name resolution requests sent by LDNS servers. When installing 3-DNS Controllers in your network, the third and final phase is to configure load balancing modes.

This chapter first describes the various load balancing modes, explains static and dynamic load balancing modes, and then describes how to configure them.

## Understanding load balancing

When the 3-DNS Controller receives a name resolution request from an LDNS, the controller 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, an **A** record for each IP address, and sends the answer back to the requesting client's LDNS server.

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

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 controller then uses a load balancing mode to choose a virtual server, within the selected pool. If the controller does not choose a virtual server in the first pool, it applies the load balancing mode to a different pool, either until it selects a virtual server, or all the pools are tried.

◆ **Load balancing within a pool**

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

Table 5.1 shows a complete list of supported load balancing modes, and indicates where you can use each mode in the 3-DNS Controller configuration. The following sections in this chapter describe how each load balancing mode works.

Load Balancing mode	Pool load balancing	Preferred	Alternate	Fallback
Completion Rate		x		x
Global Availability	x	x	x	x
Hops		x		x
Least Connections		x	x	x
Null		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

**Table 5.1** Load balancing mode usage

Load Balancing mode	Pool load balancing	Preferred	Alternate	Fallback
Static Persist		x	x	x
Topology	x	x	x	x
VS Capacity		x	x	x

*Table 5.1 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:

- Static Persist
- Round Robin
- Ratio
- Random
- Global Availability
- Topology
- Null
- Return to DNS

The Null 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.

### Static Persist mode

Static Persist mode provides static persistence of LDNS 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 will be routed through a single transaction manager (for example, a BIG-IP Controller or

other server array controller); this is beneficial for transaction-oriented traffic such as e-commerce shopping carts or online trading.

### Round Robin mode

Round Robin 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 5.1 shows a sample of the connection distribution pattern for Round Robin mode.



*Figure 5.1 Round Robin mode*

### Ratio mode

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

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

Figure 5.2 shows a sample connection distribution for Ratio mode.



*Figure 5.2 Ratio mode*

### Random mode

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

### Global Availability mode

Global Availability 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. Global Availability mode moves to the next virtual server in the list only when the current virtual server is full or otherwise unavailable. 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.

### Topology mode

Topology allows you to direct or restrict traffic flow by entering network information into a topology statement in the configuration file. This allows you to develop proximity-based load balancing. For example, client requests in a particular geographic region can be directed to servers 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 that determine proximity of client LDNS servers to the various virtual servers. 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 11, *Topology*, for detailed information about working with this and other topology features. For an example configuration using the Topology load balancing mode, see Chapter 3, *Configuring a Globally-Distributed Network*, in the **3-DNS Controller Administrator Guide**.

### Null mode

The Null 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 Null 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 Null, the 3-DNS Controller either uses the next pool, if you have multiple pools, or it 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 Null. (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 Null in each pool. If the preferred method failed, the Null mode in both the alternate and fallback methods would force the 3-DNS Controller to go to the next pool for a load balancing answer.

## Return to DNS mode

The Return to DNS mode is another special load balancing mode 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.

## 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.

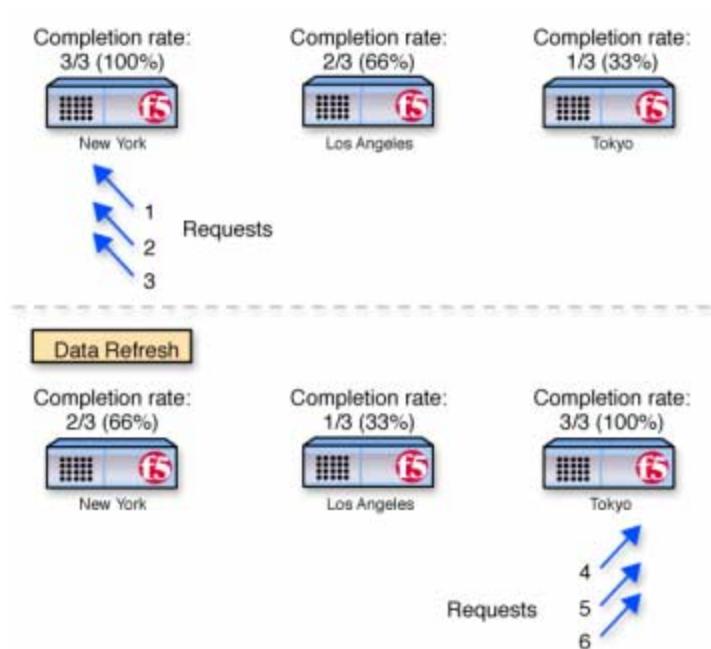
The 3-DNS Controller supports the following dynamic load balancing modes:

- Completion Rate
- Least Connections
- Packet Rate
- Round Trip Times (RTT)
- Hops
- KBPS
- Quality of Service (QOS)
- VS Capacity

## Completion Rate mode

Completion Rate mode selects a 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 5.3 shows a sample connection distribution pattern for Completion Rate mode.



**Figure 5.3** Completion Rate load balancing mode

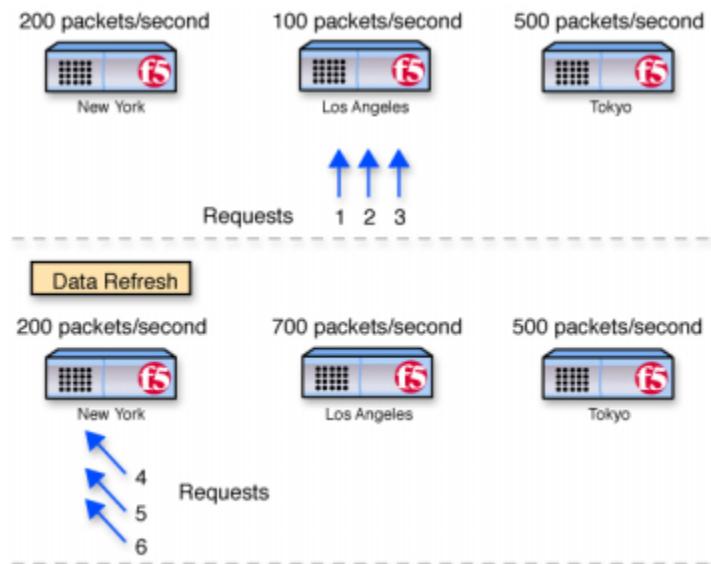
## Least Connections mode

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

## Packet Rate mode

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

Figure 5.4 shows a sample connection distribution for Packet Rate mode.

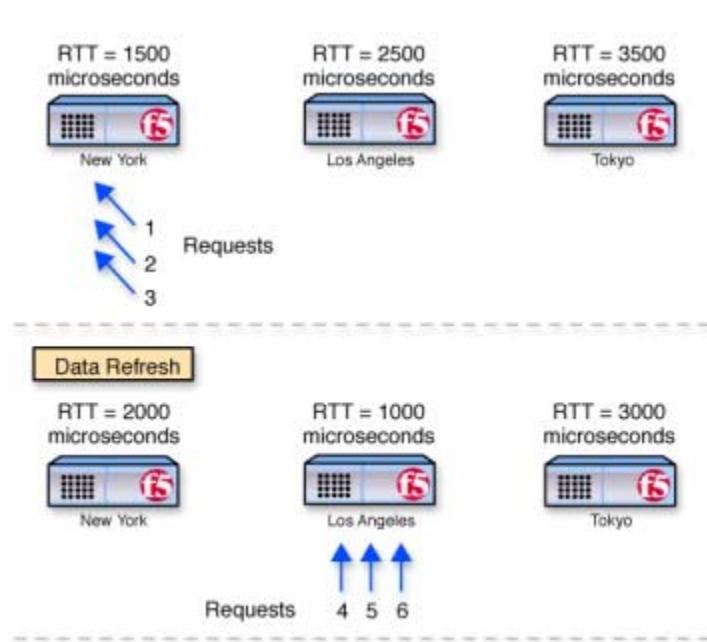


*Figure 5.4 Packet Rate mode*

## Round Trip Times mode

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

Figure 5.5 shows a sample connection distribution for Round Trip Times mode.



*Figure 5.5 Round Trip Times mode*

## Hops mode

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

## KBPS mode

KBPS mode is based on the kilobytes per second throughput rate of the virtual server. As shown in Figure 5.6, KBPS is the preferred load balancing mode for the "New York" and "Los Angeles" pools, and QOS is the preferred load balancing mode for the "Tokyo" pool. The QOS factor is 7 for KBPS.

```
wideip {
  address 192.168.101.4
  name "www.domain.com"
  port 80
  qos_coeff {
    rtt 21
    hops 0
    completion_rate 7
    packet_rate 5
    topology 1
    kbps 7
  }
  pool_lbmode ratio
  pool {
    name "New York"
    ratio 3
    preferred kbps
    address 192.168.101.5
    address 192.168.101.6
  }
  pool {
    name "Los Angeles"
    ratio 2
    preferred kbps
    address 192.168.102.5
    address 192.168.102.6
  }
  pool {
    name "Tokyo"
    ratio 1
    preferred qos
    address 192.168.103.5
    address 192.168.103.6
  }
}
```

*Figure 5.6 Example configuration using the KBPS load balancing mode*

### Quality of Service mode

Quality of Service mode uses the current performance information, calculates an overall score for each virtual server, and then distributes connections based on each virtual server's score. The performance factors that it takes into account include:

- Round trip time
- Hops
- Completion rate
- Packet rate
- Topology
- VS Capacity
- KBPS

Quality of Service mode is a customizable load balancing mode. For simple configurations, you can easily use this 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 Quality of Service mode, see *Setting up Quality of Service (QOS) mode* in Chapter 7 of the **3-DNS Controller Administrator Guide**.

### VS Capacity mode

VS Capacity mode selects the virtual server that has the most nodes **up**. If more than one virtual server has the same quantity of nodes **up**, then the 3-DNS Controller load balances using the Random mode among those virtual servers.

---

## Configuring load balancing

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

- ◆ **Global**

At the global level, you can configure default settings for the alternate and fallback load balancing modes. Then, if you don't specify alternate or fallback modes when defining a wide IP, the 3-DNS Controller uses the alternate and fallback modes you have configured at the global level. You can find instructions on how to configure global alternate and fallback modes in *Setting global alternate and fallback modes*, on page 5-21.

- ◆ **Wide IP**

When defining a wide IP, if you have multiple pools in your wide IP, you first specify which load balancing mode to use in selecting the pool in the wide IP. Next, you specify which preferred, alternate, and fallback load balancing modes to use in selecting the virtual server within the selected pool. You can find instructions on how to configure these load balancing modes in the section, *Adding a wide IP*, on page 5-14.

## Understanding wide IPs

After you configure the BIG-IP Controllers, EDGE-FX Caches, hosts, and the virtual servers they manage, you need to group the configured virtual servers into a wide IP. A **wide IP** is a mapping of a fully-qualified domain name (FQDN) to a set of virtual servers that host the domain 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 Controller, 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**

---

*NameSurfer, an 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 it (available from the Configuration utility).*

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 *Setting up load balancing for services that require multiple ports*, in Chapter 7 of the **3-DNS Controller Administrator Guide**.

## Understanding pools

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

## Adding 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 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. For help on defining wide IPs, click **Help** on the toolbar.  
The wide IP is added to your configuration.

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

### **To define a wide IP using the command line utility**

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

Figure 5.7 shows the correct syntax for the **wideip** statement.

```

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

```

**Figure 5.7** Syntax for the **wideip** statement

Figure 5.8 shows a sample **wideip** statement. This statement defines a wide IP named **mx.wip.domain.com**, with an alias of **mail.wip.domain.com**. The wide IP contains two pools, with **pool\_1** receiving three times as many requests as **pool\_2**. The 3-DNS 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 controller 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.domain.com"
  alias        "mail.wip.domain.com"
  pool_lbmode  ratio
  pool {
    name       "pool_1"
    ratio      3
    preferred  rtt
    alternate  random
    address    192.168.101.50
    address    192.168.102.50
    address    192.168.103.50
  }
  pool {
    name       "pool_2"
    ratio      1
    preferred  ratio
    address    192.168.104.50  ratio 2
    address    192.168.105.50  ratio 1
  }
}
```

**Figure 5.8** Example syntax for defining a wide IP

### Using the LDNS round robin wide IP attribute

LDNS round robin is an attribute that you can use in conjunction with any load balancing mode. The LDNS round robin attribute allows the 3-DNS 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 when 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 all other configured pools

When your network includes cache appliances hosting content from an origin site, you can designate the origin site as the last resort pool to handle requests if your cache virtual servers have reached their thresholds. You can also use the last resort pool to designate an overflow network so your origin servers remain available if network traffic spikes. You can only designate one last resort pool within a wide IP.

#### **To designate a last resort pool using the Configuration utility**

1. In the navigation pane, select **Wide IPs**.  
The Wide IP List screen opens.
2. From the Pools column, select the pools for the wide IP for which you want to create a last resort pool.  
The Modify Wide IP Pools screen opens.

3. From the Pool Name column, click the pool that you want to designate as the last resort pool.  
The Modify Load Balancing for [pool name] screen opens.
4. Check the box next to **Last Resort Pool**, and click **Update**.

### To designate a last resort pool using 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 5.9 shows an example of a last resort pool definition.

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

*Figure 5.9 Example of a last resort pool definition*

## 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 manually.

### ◆ 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 them, 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 manually configure wide IPs, use the **3dparse** utility to verify **wideip.conf** syntax before you start **named**. To use this utility, type **3dparse** on the command line. For details on the **3dparse** utility, see the **3dparse** man page.

◆ **/var/log/messages**

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

◆ **BIND syntax**

If you are setting up the configuration manually, 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>

## 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 modes
- 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 modes

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

### To configure global alternate and fallback load balancing modes 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 mode fail.
4. In the **Default Fallback** box, specify the load balancing mode to use should the preferred and alternate modes fail. If all modes fail, requests are returned to DNS.
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 manually configure global alternate and fallback load balancing modes

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.

2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate or add the **globals** statement. The **globals** statement should be at the top of the file.
4. Use the syntax shown in Figure 5.10 to define global alternate and fallback load balancing modes.

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

**Figure 5.10** Configuring global alternate and fallback load balancing modes

Figure 5.11 shows a sample **globals** statement that defines global load balancing variables.

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

**Figure 5.11** Sample syntax for setting global load balancing variables

## Understanding TTL and timer values

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

Table 5.2 describes each TTL value, as well as its default setting.

Parameter	Description	Default
Server TTL	Specifies the number of seconds that the 3-DNS Controller uses BIG-IP Controller 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
Virtual server TTL	Specifies the number of seconds that the 3-DNS Controller uses virtual server information (data acquired from a BIG-IP Controller, EDGE-FX Cache, or other host machine about a virtual server) for name resolution and load balancing.	120
Trace 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 the wide IP <b>A</b> 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 <b>default_ttl</b> value.	30

**Table 5.2** *TTL values and default settings*

Each 3-DNS object also has a timer value. A timer value defines the frequency (measured in seconds) at which the 3-DNS 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 5.3 describes each timer value, as well as its default setting.

Parameter	Description	Default
Server data refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller refreshes BIG-IP Controller and EDGE-FX Cache 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
Trace 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
Remote nodes query	Specifies the frequency (in seconds) at which the 3-DNS Controller queries remote 3-DNS Controllers and BIG-IP Controllers.	60

**Table 5.3** Time values and default settings

Parameter	Description	Default
3-DNS Sync Time Tolerance	<p>Specifies the number of seconds that one 3-DNS Controller's time setting is allowed to be out of sync with another 3-DNS Controller's time setting. If the difference between the times on the controllers is higher than the time tolerance setting, the time setting on the controller running behind is reset to match the controller with the most recent time. For example, if the time tolerance is 5 seconds, and one 3-DNS Controller is running 10 seconds ahead of the other, the controller running behind has its time reset to match the one running 10 seconds ahead. If the second controller were running only 2 seconds ahead of the other, the time settings would remain unchanged.</p> <p><b>Note:</b> 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 <b>10</b>. In the event that NTP fails, the 3-DNS Controller uses the <b>time_tolerance</b> variable to maintain synchronization.</p>	10
Timer Sync State	<p>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). The first enabled 3-DNS Controller listed in a sync list is the principal, and the others are receivers. The controller changes states under the following circumstances: if the principal is disabled, the next enabled controller listed in the sync list becomes the principal. When the original principal becomes enabled, it once again becomes principal, and the temporary principal returns to a receiver state.</p>	30
Persist Cache	<p>Specifies the interval (in seconds) at which the 3-DNS Controller archives the paths and metrics data.</p>	3600

**Table 5.3** Time values and default settings

### To configure global TTL and timer values using the Configuration utility

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

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

### To manually configure global TTL and timer values

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, choose **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate or add the **globals** statement. The **globals** statement should be at the top of the file.

4. Use the syntax shown in Figure 5.12 to define global TTL and timer values.

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

**Figure 5.12** *Syntax for configuring global TTL and timer values*



# 6

---

---

## Network Map

---

---

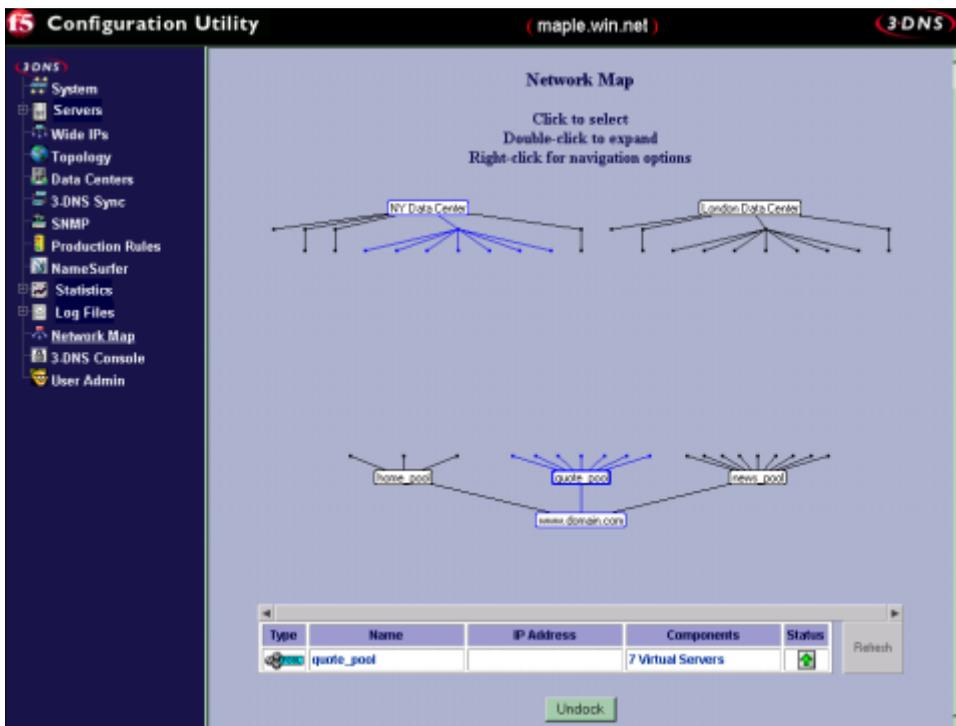




## Network Map

The 3-DNS Controller 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 Controller network
- 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 Controller in the data center, **NY Data Center**.

## Working with the Network Map

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

### Viewing the Network Map

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

#### To view the Network Map using the Configuration utility

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

### Using the Network Map to review and modify the network configuration

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

IPs that contain a virtual server that belongs to the servers in the selected data center are also highlighted. You can continue to double-click the objects to narrow your scope.

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

## Using the information table on the Network Map

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

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

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



# 7

---

---

## 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 QOS 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 7-8, which describes the scripting language you use to configure production rules manually. You may also want to contact a technical support engineer for additional assistance with complex configurations.

## Setting up production rules in the Configuration utility

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

- ◆ **Define the type of rule**

There are two types of rules: global production rules and wide IP production rules.

- ◆ **Define the rule trigger**

There are two types of rule triggers: a set time or time interval, and specific system events.

- ◆ **Define the action taken**

There are two basic types of rule actions: sending user-definable messages to log files or email accounts, and changing specific load balancing settings.

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

## Viewing, adding, and deleting production rules

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

Note that you can modify existing production rules by clicking the rule name in the list, and you can delete a production rule at any time by clicking the Delete button (trash can icon) 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 the following section.

**◆ Wide IP production rules**

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

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

## Defining time-based triggers

The next step in the wizard prompts you to choose a trigger for the production rule. There are two basic types of triggers that you can set up: time-based triggers and event-based triggers. This section describes the options for the time-based triggers, and the following section describes options for the event-based triggers. Once you review the information for the type of trigger you want to set up, you can skip to the section about choosing an action on page 7-7.

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

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

**To apply a time of day variable**

1. From the Time Variable table, select **Time**.
2. From the **Start Time, Hour** box, select the hour you want the production rule action to begin.
3. From the **Start Time, Minutes** box, select the minute you want the production rule action to begin.

4. From the **Stop Time, Hour** box, select the hour you want the production rule action to stop.
5. From the **Stop Time, Minutes** box, select the minute you want the production rule action to stop.

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

### **To apply a day of the week variable**

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

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

### **To apply a date variable**

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

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

### To apply a combined date and time variable

1. From the Time Variable table, select **Date/Time**.  
Two tables open and you select the start and stop dates and times.
2. In the **Start Date** box, type the date you want the production rule action to begin (mm/dd/yyyy).
3. In the **Stop Date** box, type the date you want the production rule action to stop (mm/dd/yyyy).
4. From the **Start Time, Hour** box, select the hour you want the production rule action to begin.
5. From the **Start Time, Minutes** box, select the minute you want the production rule action to begin.
6. From the **Stop Time, Hour** box, select the hour you want the production rule action to stop.
7. From the **Stop Time, Minutes** box, select the minute you want the production rule action to stop.

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

## Defining event-based triggers

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

Standard events that can trigger both global and wide IP production rules include:

- ◆ **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 Controller, BIG-IP Controller, 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 Controller configuration is re-read (for example, when an **ndc reload** command is issued).

## Choosing the action

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.

- ◆ **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 return 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 a Professional Services representative 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 Controller configuration, you must store manually configured production rules 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.

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

### Executing and managing production rules

The **3dscript** utility manages and executes production rules according to the following guidelines:

- **3dscrip**t 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.
- **3dscrip**t supports event-driven execution of production rules using the **when** statement. You can use the **when** statement only in global production rules.
- **3dscrip**t supports periodic execution of production rules using the **every** statement. You can use the **every** statement only in global production rules.
- 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 if statement

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

- The **if** statement can be specified in the scope of a wide IP **pool** statement.
- The **if** statement can be nested in another **if** statement.
- Multiple **if** statements can be specified in the same scope.
- The nesting of **if** statements is unlimited except by the memory capacity of the 3-DNS Controller.
- The first form of an **if** statement is:

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

- The second form of an **if** statement is:

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

- The conditional-expression is composed of one of these:

- 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
- The primitive-expression can be one of these:
  - 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
- A logical-operator is either:
  - `||` (logical OR)
  - `&&` (logical AND)
- A relational-operator is one of these:
  - `==` (equality)
  - `!=` (not equal)
  - `>` (greater than)
  - `>=` (greater than or equal to)
  - `<` (less than)
  - `<=` (less than or equal to)
- An arithmetic-operator is:
  - `mod` (modulus)
- A unary operator is either:
  - `!` (unary negation)
  - (unary minus)
- A keyword is one of the following:
  - `day`

- time
- date
- datetime
- ldns\_ip
- wip\_ip
- wip\_name
- wip\_num\_resolves
- preferred
- alternate
- fallback
- rtt
- completion\_rate
- hops
- packet\_rate
- topology
- An intrinsic function is either:
  - isLdnsInNet(ip, mask)
  - isLdnsInAS(ip, mask)
- The precedence of logical, relational, and unary operators is the same as in ANSI-c.

## The when statement

The **when** statement is a standard statement which 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 **wideip.conf**, 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 form of a **when** statement is:  
`when(event) { <action> ... }`
- An event can be one of the following (see page 7-5 for detailed descriptions of each event):
  - ResolveNameBegin
  - ResolveNameEnd
  - FallbackToStatic
  - SIGINT
  - SIGHUP
  - SIGUSR1
  - SIGUSR2
  - SIGCHLD
  - ReapPaths
  - ReapLdns
  - CRC\_Failure
  - DownServer
  - DownVS
  - DoneInit
  - DoneConfigFile

### The every statement

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

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

- The form of the **every** statement is:

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

## 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.

- ◆ **preferred <lbmode>**  
This action changes the preferred load balancing method in a wide IP. You can use this action only in a wide IP production rule.
- ◆ **alternate <lbmode>**  
This action changes the alternate load balancing method in a wide IP. You can use it only in a wide IP production rule.
- ◆ **fallback <lbmode>**  
This action changes the fallback load balancing method in a wide IP. You can use this action only in a wide IP production rule.
- ◆ **log(<string>)**  
This action sends the specified string to the syslog utility, which writes the string to the syslog file. You can use this action in either a wide IP production rule or a 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). You can use this action in either a wide IP production rule or a global production rule.

- ◆ **vs(<ip>:<port>).ratio <n>**  
This action changes the ratio setting for a specific virtual server in a wide IP pool. You can use this action only in a wide IP production rule.
- ◆ **return\_vs(<ip:port>)**  
This action skips the load balancing process and instead returns the specified virtual server to the requesting client. You can use this action only in a wide IP production rule.

## 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 LDNS
- Hacker detection

### Load balancing 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 7.1 shows a sample production rule that handles the connections according to the anticipated load at specific times of the day.

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

**Figure 7.1** Load balancing by time of day

## Load balancing according to LDNS

One interesting application of production rules is when you create a rule that triggers based on a specific LDNS server making 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 7.2 uses the address of the requesting LDNS to determine which virtual server should receive the connection.

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

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

**Figure 7.2** Load balancing by IP address of LDNS

## 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 7.3 sends the hacker to a special server, rather than flat out rejecting the connection. As an alternative, you could 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.
    // It's reserved for us to watch hackers under the microscope.
    log2mail("Hacker $ldns_ip came back")
    return_vs(192.168.1.46:80)
  }
}
```

**Figure 7.3** *Sending a hacker to a specific server*

A related example, shown in Figure 7.4, illustrates a production rule that deals with attacks against iQuery communications. The production rule would warn you if the 3-DNS Controller detected a hack attempt against the iQuery protocol, based on a communication failure.

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

**Figure 7.4** *Detecting an iQuery failure due to potential attack*



# 8

---

---

## Resource Records

---

---





## Resource records

A *resource record* (RR) consists of a name, a type, and data that is specific to the type. These resource records, in a hierarchical structure, make up the domain name service (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 the previous resource record.

◆ **ttl**

The second field, **ttl** (time to live), is optional. This field specifies how long this data is stored by the LDNS. If this field is left blank, the default time to live 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".

◆ **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.

## Types of resource records

There are many types of resource records currently in use. This section provides an overview of the most common resource record types, and lists other types of resource records. There are six standard types of resource records:

Type	Description
A (Address)	Converts 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 a domain's name servers.
PTR (Pointer)	Converts IP addresses to host names.
SOA (Start of Authority)	Marks the beginning of a zone's data, defines default parameters for a zone.

**Table 8.1** Standard resource records

### A (Address)

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

Figure 8.1 shows an example of an **A** record:

{name}	{ttl}	addr-class	A	address
ucbarpa		IN	A	128.32.0.4
		IN	A	10.0.0.78

**Figure 8.1** 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 8.2 shows an example of a CNAME resource record:

alias	{ttl}	addr-class	CNAME	Canonical name
ucbmonet		IN	CNAME	monet

*Figure 8.2 Example of a CNAME record*

## MX (Mail Exchange)

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

Figure 8.3 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 8.3 Example of an MX record*

## NS (Name Server)

The Name Server resource record, **NS**, defines the name server(s) for a given domain, creating a delegation point and a subzone. The first name field specifies the zone that is serviced by the name server that is specified by the second name. Every zone needs at least two name servers.

Figure 8.4 shows an example of an **NS** resource record:

```
{name}   {ttl}   addr-class  NS   Name servers name
                IN      NS   ucbarpa.Berkeley.Edu.
```

*Figure 8.4 Example of an NS record*

### PTR (Pointer)

A Name Pointer 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 8.5 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        PTR   monet.Berkeley.Edu.
```

*Figure 8.5 Example of a PTR record*

### SOA (Start of Authority)

The Start of Authority, **SOA**, record starts every zone file. There must be exactly one **SOA** record per zone.

The following is an example of an **SOA** resource record:

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

*Figure 8.6 Example of an SOA record*

The **SOA** record-specific fields are defined as follows:

- ◆ **Person in charge**  
The email address for the person responsible for the name server, with at (@) changed to a dot (.) .
- ◆ **Serial number**  
The version number of this 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, in seconds, between calls 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 8.2 lists less common resource record types. For more information on these, see RFCs 1035, 1183, and 1664.

Type	Description
AAAA	IPv6 address
AFSDB	AFS database location
GPOS	Geographical position
HINFO	Host information

*Table 8.2 Other types of resource records*

---

Type	Description
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 8.2** *Other types of resource records*

# 9

---

---

## Scripts

---

---





---

## Scripts

This chapter provides information about how each script that is shipped with the 3-DNS Controller works. If you plan on doing a scripted task manually, you should find this section helpful. Many scripts correspond to commands on the 3-DNS Maintenance menu.

◆ **Note**

---

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

### 3dns\_admin\_start

The **3dns\_admin\_start** script corresponds to the **Restart 3-DNS Web Administration** command on the 3-DNS Maintenance menu. This command restarts the 3-DNS web server.

### 3dns\_auth

The **3dns\_auth** script corresponds to the **Generate RSA Authentication** command on the 3-DNS Maintenance menu. All 3-DNS Controller scripts are easier to use when you generate password authentication. Any time you add a new 3-DNS Controller or BIG-IP Controller to a network, you can run the **3dns\_auth** script, and if no **ssh** key exists on the controller, the script will configure **ssh** access.

◆ **Note**

---

*This script is not available in the non-crypto version of the 3-DNS Controller.*

### 3dns\_dump

The **3dns\_dump** script saves the current state of the **named** cache to a new **/var/3nds/etc/wideip.conf** file.

## 3dns\_sync\_metrics

The **3dns\_sync\_metrics** script corresponds to the **Synchronize Metrics Data** command on the 3-DNS Maintenance menu. You should use this script only when you are configuring a new 3-DNS Controller. This script prompts you to copy metrics data from a remote 3-DNS Controller to the local 3-DNS Controller.

## 3dns\_web\_config

The **3dns\_web\_config** script corresponds to the **Reconfigure 3-DNS Web Administration** command on the 3-DNS Maintenance menu. This script lets you make configuration changes to the 3-DNS web server.

## 3dns\_web\_passwd

The **3dns\_web\_passwd** script corresponds to the **Change/Add Users for 3-DNS Web Administration** command on the 3-DNS Maintenance menu. This script secures the 3-DNS web server using basic authentication. This script lets you provide restricted or administrative access to the 3-DNS web server for selected users only, and assigns passwords for those users. Users with restricted access have access to the statistics area only. Users with administrative access have access to all areas of the 3-DNS web server.

◆ **Note**

---

*The **3dns\_web\_passwd** script is run by the First-Time Boot utility. You can run this script again any time you need to provide access for another user.*

## 3dnsmaint

The **3dnsmaint** script opens the 3-DNS Maintenance menu.

---

## 3dprint

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

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

- ◆ **Wide IPs**

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

Displays statistics about the globals sub-statements; the statistics include such things as the current and default values for each of the globals sub-statements, and whether you have to restart **named** when you make changes to the parameters.
- ◆ **Summary**

Displays summary statistics, such as the 3-DNS Controller version, the total number of resolved requests, and the load balancing methods used to resolve requests.
- ◆ **Data Centers**

Displays statistics about the data centers, and their servers, in your network. The statistics include such things as the names of the data centers, the name or IP address of the servers in the data center, and whether the data center is enabled or disabled.
- ◆ **Sync Groups**

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

### 3ndc

The **3ndc** script starts the **3ndc** utility, which is described in the **3ndc** man page. Note that **ndc** is an alias for **3ndc**.

---

## big3d\_check

The **big3d\_check** script corresponds to the **Check big3d** command on the 3-DNS Maintenance menu. This script checks that each BIG-IP Controller listed in the **bigips.txt** file is running the **big3d** agent.

## big3d\_install

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

**big3d\_install** performs the following procedure on each BIG-IP Controller or EDGE-FX Cache:

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

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

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

## big3d\_restart

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 Controller.

## big3d\_version

The **big3d\_version** script corresponds to the **Check versions of named, BIG-IP kernel and needed big3d** command on the 3-DNS Maintenance menu. This script displays version numbers for all BIG-IP Controllers known to the 3-DNS Controller, as well as the version numbers of the **big3d** agent and **named** utility running on each BIG-IP Controller.

## edit\_lock

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

To use this script, type the following:

```
edit_lock <file name>
```

## edit\_wideip

The **edit\_wideip** script corresponds to the **Edit 3-DNS Configuration** command on the 3-DNS Maintenance menu. This script opens the **wideip.conf** file for editing, copies it to all other 3-DNS Controllers in the local 3-DNS Controller's sync group, and restarts **named**.

---

## install\_key and F5makekey

The **install\_key** script corresponds to the **Generate and Copy F5 iQuery Encryption Key** command on the 3-DNS Maintenance menu. This script starts the **F5makekey** script and generates a seed key for encrypting communications between the 3-DNS Controllers and (if you have any in your network) BIG-IP Controllers. The **install\_key** script creates and distributes the iQuery key to all BIG-IP Controllers and other 3-DNS Controllers on your network.

### ◆ Note

---

*This script is not available in the non-crypto version of 3-DNS Controller.*

To start the **F5makekey** script, type the following from **/usr/contrib/bin**:

```
f5makekey
```

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

## syncd\_checkpoint

The **syncd\_checkpoint** script corresponds to the **Checkpoint synced files** command on the 3-DNS Maintenance menu. This script creates a **checkpoint file**. A checkpoint file is a compressed tar file that contains an archive of the files that are synchronized.

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

```
/var/3dns/staging/checkpoint/default.tar.gz
```

### ◆ Note

---

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

The **syncd\_checkpoint** script can take the following optional arguments:

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

The options for **syncd\_checkpoint** are defined as follows:

**-c <name>**

Creates a checkpoint file with the specified file name. You can also specify a non-default path for the file, unless the path starts with a slash (/). The default path for checkpoint files is **/var/3dns/staging/checkpoint/**. The **syncd\_checkpoint** script automatically appends a **.tar.gz** extension to the end of the file name.

**-i**

Runs the script in an interactive session, which means that you are prompted for a file name.

## syncd\_rollback

The **syncd\_rollback** script corresponds to the **Rollback checkpoint** command on the 3-DNS Maintenance menu. This script unrolls a checkpoint file, which contains an archive of all synchronized files. This has the effect of replacing the current files with the files archived in the checkpoint file.

The **syncd\_rollback** script can take the following optional arguments:

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

The options for **syncd\_rollback** are defined as follows:

**-c**

Unrolls the most recently created checkpoint file, whether it is in the default location or elsewhere.

**-c <name>**

---

Unrolls the specified checkpoint file, whether it is in the default location or elsewhere. It is not necessary to end the name with **.tar.gz**, as this suffix is assumed.

**-r**

Restores archived files with their old timestamps. This means that if any of the synchronized files were updated on a remote 3-DNS Controller, the updated files will overwrite any older files contained in the checkpoint file.

**-u**

Restores archived files with updated timestamps with the current time. This means that the files in the checkpoint are synchronized to the remote 3-DNS Controllers and overwrite the existing files on the remote 3-DNS Controllers.

**-i**

Runs the script in an interactive session, which means that you are prompted for option information.

◆ **Note**

---

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

## syncd\_start

The **syncd\_start** script corresponds to the **Restart syncd** command on the 3-DNS Maintenance menu. This script restarts the **syncd** daemon if it is already running, or starts it if it is not.

You can run this script with or without arguments. If you run **syncd\_start** without specifying arguments, the script starts or restarts **syncd**.

The **syncd\_start** script can take the following optional arguments:  
**syncd\_start** [-c] [-c <name>] [-r] [-u] [-i]

The options for **syncd\_start** are defined as follows:

**-c**

Before restarting **syncd**, unrolls the most recently created checkpoint file, whether it is in the default location or elsewhere.

**-c <name>**

Before restarting **syncd**, unrolls the specified checkpoint file, whether it is in the default location or elsewhere. It is not necessary to end the name with **.tar.gz**, as this suffix is assumed.

**-r**

Restores the archived files with their old timestamps. This means that if any of the synchronized files were updated on a remote 3-DNS Controller, the updated files overwrite the rolled back files.

**-u**

Restores the archived files with updated timestamps to the current time. This means that the files in the checkpoint file overwrite any updated files on remote 3-DNS Controllers.

**-i**

Runs the script in an interactive session, which means that you are prompted for option information.

◆ **Note**

---

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

## syncd\_stop

The **syncd\_stop** script corresponds to the **Stop syncd** command on the 3-DNS Maintenance menu. This script stops the **syncd** daemon if it is running.

You can run this script with or without arguments. If you run **syncd\_stop** without specifying arguments, the script simply stops **syncd**.

The **syncd\_stop** script can take the following optional arguments:

**syncd\_stop [-c] [-c <name>] [ -i]**

---

The options for **syncd\_stop** are defined as follows:

**-c**

Creates a checkpoint file in the default location before stopping **syncd**.

**-c name**

Creates a checkpoint file with the specified name and path before stopping **syncd**.

**-i**

Runs the script in an interactive session, which means that you are prompted for option information.



# 10

---

---

## SNMP

---

---

- Working with SNMP on the 3-DNS Controller
  - Configuring SNMP on the 3-DNS Controller
  - Downloading the MIBs
  - Understanding configuration file requirements
  - Configuring options for the checktrap script
  - Configuring the 3-DNS SNMP agent using the Configuration utility
  - Configuring host SNMP settings on the 3-DNS Controller
  - Configuring SNMP agents on hosts
-



---

## Working with SNMP on the 3-DNS Controller

This chapter describes the management and configuration tasks for the simple network management protocol (SNMP) agent and management information bases (MIBs) available with the 3-DNS Controller.

### **WARNING**

*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.*

The 3-DNS SNMP agent and MIBs allow you to monitor the 3-DNS Controller by configuring traps for the SNMP agent or by polling the controller with your standard network management station (NMS). The 3-DNS SNMP agent has the following options to ensure secure management:

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

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

## 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/snmptrap.conf
- /etc/syslog.conf
- Configure options for the checktrap script

## Downloading the MIBs

The 3-DNS Controller includes a private 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 three MIB files from the 3-DNS directory `/usr/contrib/f5/mibs`, or you can download the files from the **Additional Software Downloads** section of the Configuration utility home page. The files you need are:

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

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

---

## Understanding configuration file requirements

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

### /etc/hosts.deny

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

```
ALL : ALL
```

### /etc/hosts.allow

#### ◆ Note

---

*If you prefer, instead of modifying this file manually, you can use the Configuration utility to specify the hosts that are allowed to access the SNMP agent. See the section titled, **To set SNMP properties using the Configuration utility**, on page 10-9.*

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.

For a specific list of addresses, type in the list of addresses you want to allow access to the SNMP agent. Addresses in the list must be separated by blank space or by commas. The basic syntax is as follows:

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

For example, if you type the following line, the SNMP agent accepts connections from the specified IP addresses:

```
snmpd: 128.95.46.5 128.95.46.6 128.95.46.7
```

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

```
daemon: NETWORKADDRESS/MASK
```

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

```
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.

## /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.

### ◆ Note

---

*If you prefer, instead of modifying this file manually, you can use the Configuration utility to set these SNMP properties. See the section titled, **To set SNMP properties using the Configuration utility**, on page 10-9.*

A few of the necessary variables are listed below:

#### ◆ System Contact Name

The System Contact is a MIB-II simple string variable defined by almost all SNMP boxes. 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 boxes. It is a simple string that defines the location of the box. 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>** 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)**.
  - **authtrappable <integer>**

Set this variable to **1** so that traps can be sent for authentication warnings. Set the variable to **2** to disable it.  
*Note: To change the trap port, be sure the **trapport** line precedes the **trapsink** line. If you use more than one **trapport** line, there must be one **trapport** line before each **trapsink** line. The same is true for **trapcommunity**; if you use more than one **trapcommunity** line, there must be one **trapcommunity** line before each **trapsink** line.*
- ◆ **System IP Setting**

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

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

## /etc/snmptrap.conf

The configuration in **/etc/snmptrap.conf** 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 10.1

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

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

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

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

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

**Figure 10.1** Excerpt from the */etc/snmptrap.conf* file

Some of the OIDs have been permanently mapped to specific 3-DNS Controller 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 red to green
- Pool green to red
- 3-DNS Controller active to standby

- 3-DNS Controller standby to active

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

## `/etc/syslog.conf`

To generate traps, you must configure **syslog** to send syslog lines to **checktrap.pl**. If the syslog lines match the specified regular expression in the **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 **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 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 the regular expression, an SNMP trap is sent.

### Options for checktrap

```
snmpd_conf_file=<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`.

```
trapd_conf_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/snmptrap.conf**.

**trap\_program=<snmp trap program>**

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

**no\_date\_strip**

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

**usage**

This prints a usage string.

## Configuring the 3-DNS SNMP agent using the Configuration utility

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

- ◆ **Client access**  
You can define a network address and netmask for a workstation from which SNMP requests are acceptable.
- ◆ **System information**  
You can name a system contact, a machine location, and a community string.
- ◆ **Trap configuration**  
You can enter a trap sink and a trap community.

### To set **SNMP** properties using the **Configuration utility**

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

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

## Configuring host SNMP settings on the 3-DNS Controller

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

The SNMP prober collects the following information:

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

The 3-DNS Controller uses the packet rate information for load balancing. The information is displayed in the Host Statistics screen in the Configuration utility for your convenience.

### To configure host SNMP settings using the Configuration utility

1. In the navigation pane, expand the **Servers** item, and click **Host Servers**.
2. From the Host Server 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 host SNMP settings. For help on configuring the host SNMP settings, click **Help** on the toolbar.

### To configure host SNMP settings using the command line utility

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

Figure 10.2 shows the SNMP syntax in bold.

```
server {
  type host
  address <IP address>
  name <"host_name">
  probe_protocol <tcp | icmp>
  [ prober <IP address> ]
  port <port number> | service <"service name">
  [ snmp {
    agent <generic | ucd | solstice | ntserver | ciscold2 | ciscold3>
    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 <tcp | icmp> ]
  }
}
```

*Figure 10.2* Configuring host SNMP settings

## Configuring SNMP agents on hosts

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



||

---

---

# Topology

---

---





---

# Topology

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 Controller includes a database that maps IP addresses to geographic locations. With this database, the controller can use the geographic attributes of local DNS servers to direct traffic.

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

## Setting up topology records

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

- IP subnet (CIDR definition)
- Wide IP pool (managed by the 3-DNS Controller)
- Data center (managed by the 3-DNS Controller)
- Country (based on top-level domain codes, as specified by IANA, the Internet Assigned Numbers Authority)
- Continent
- America Online (AOL) (for LDNS server location endpoints only)

The relative weight, or 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 that value.

A *topology statement* is composed of one or more topology records. Figure 11.1 is an example of a topology statement, with two topology records, as it appears in the Configuration utility.

//virtual server	LDNS	Score
pool."origin"	continent."North America"	100
pool."cache_farm"	!continent."North America"	100

**Figure 11.1** Example of a topology statement

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

## 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 virtual servers in specific data centers within your network or to content delivery networks.

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

1. In the navigation pane, click **Wide IPs**.  
The Wide IP List screen opens.

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

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

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

Figure 11.2 shows a sample wide IP definition where topology is the load balancing mode for the pools in this wide IP configuration.

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

  pool {
    name      "cache_farm"
    fallback  null
    address   192.168.44.1
    address   192.168.44.2
  }

  pool {
    name      "origin"
    address   172.168.11.1
    address   172.168.11.2
  }
}
```

*Figure 11.2 Example syntax for a wide IP that uses Topology pool load balancing*

### Using the Topology load balancing mode within 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 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 within a pool from the command line

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

The example in Figure 11.3 shows a sample wide IP definition where **topology** is the load balancing mode within a pool.

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

*Figure 11.3 Example of Topology load balancing mode within a pool*

The following variables are allowed in the **topology** statement to specify pools, data centers, continents, and countries, in addition to the traditional CIDR blocks, for both servers and local DNS servers.

Variable	Description
pool	Specify a wide-IP pool to score for load balancing. Note that pool names can be duplicated across wide IPs
datacenter	Specify a data center to score for load balancing.
continent	Specify one of these continents for load balancing: <b>"North America"</b> , <b>"South America"</b> , <b>"Europe"</b> , <b>"Asia"</b> , <b>"Australia"</b> , <b>"Africa"</b> , or <b>"Antarctica"</b> .
country	Specify a country for load balancing using one of the two-letter country codes found in the file <b>/var/3dns/include/net.ccdb</b> .
isp.AOL	For local DNS servers only, specify the Internet service provider, America Online (AOL).

**Table 11.1** Variables used in the **topology** statement

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

```

topology {
// server          ldns          score
"pool.origin"     cont."North America" 100
"pool.cache_farm" !cont."North America" 100
}

```

**Figure 11.4** 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 11.4.

# 12

---

---

## Utilities

---

---





---

## Utilities

The 3-DNS Controller includes several utilities. These utilities allow you to configure the DNS and the various features of the 3-DNS Controller. You may also want to review Chapter 9, *Scripts*, for additional 3-DNS Controller configuration options.

### 3-DNS Controller utilities documentation

You can access the most current documentation on 3-DNS Controller utilities by using the Configuration utility or by using the command line.

#### **To access documentation on 3-DNS Controller utilities using the Configuration utility**

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

#### **To access documentation on 3-DNS Controller utilities from the command line**

Using the command line, you can display a list of utilities that fall into a particular category, or else display the man page for a specific utility.

#### **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 which 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 Controller 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
```

---

---

# Index

---

---





/etc/hosts.allow file 10-3  
/etc/snmptrap.conf file 10-6  
/etc/syslog.conf file 10-7  
3-DNS Controller  
    viewing network structure 6-1  
3-DNS guides, types of 1-1  
3dns\_admin\_start 9-1  
3dns\_auth 9-1  
3dns\_dump 9-1  
3dns\_sync\_metrics 9-2  
3dns\_web\_config 9-2  
3dns\_web\_passwd 9-2  
3dnsmaint 9-2  
3dprint 9-3  
3dscript  
    managing production rules 7-8  
3ndc 9-4

## A

A resource records 8-2  
actions supported by production rules 7-13  
Ask F5 knowledge base 1-4

## B

big3d agents 3-2  
    and factory types 3-3  
    configuring 3-1  
    installing 9-5  
    installing on BIG-IP Controllers 3-1, 3-3  
    installing on EDGE-FX Caches 3-3  
big3d\_check 9-5  
big3d\_install 9-5  
big3d\_restart 9-6  
big3d\_version 9-6  
BIG-IP Controller  
    installing big3d agent 3-3

## C

checkpoints 9-7  
CNAME resource records 8-3  
command line conventions 1-3

completion rate  
    load balancing mode 5-8  
Configuration utility  
    setting up production rules 7-1  
custom production rules  
    assistance 7-8  
    examples 7-14

## D

date variable  
    applying 7-4  
day of the week variable  
    applying 7-4  
defining 7-3

## E

ECV 4-1  
ECV service monitors 4-1  
EDGE-FX Cache  
    installing big3d agent 3-3  
edit\_lock 9-6  
edit\_wideip 9-6  
encryption  
    using f5makekey 9-7  
event-based triggers  
    defining 7-5  
every statement  
    guidelines 7-12  
    in production rules 7-12  
examples  
    Topology mode 11-5  
Extended Content Verification 4-1

## F

f5makekey 9-7  
factories  
    default settings 3-4  
factories, types of  
    discovery 3-3  
    hops 3-3  
    nonconfigurable 3-3  
    probing 3-3

- SNMP 3-3
- file monitors 4-1
- firewalls
  - configuring for 3-9

## G

- global production rules 7-2
- global variables
  - configuring load balancing 5-20
- globals statement
  - load balancing variables 5-22

## H

- hacker detection 7-16
- help, finding 1-3

## I

- if statement
  - guidelines 7-9
  - in production rules 7-9
- Install and Start big3d command 3-3
- install\_key 9-7
- iQuery
  - configuring for firewalls 3-9

## K

- key
  - generating for encryption 9-7

## L

- last resort pool
  - about 5-18
  - configuring 5-18
- LDNS
  - load balancing 7-16
- LDNS round robin
  - about 5-18
- least connections
  - load balancing mode 5-8
- load balancing

- configuring global variables 5-20
  - using production rules 7-14, 7-16
- load balancing according to LDNS 7-16
- load balancing modes
  - completion rate 5-8
  - least connections 5-8
  - null 5-6
  - packet rate 5-9
  - random 5-5
  - ratio 5-4
  - return to DNS 5-7
  - round robin 5-4
  - round trip times 5-9
  - static persist 5-3

## M

- man pages 12-1
- management tool
  - production rules 7-1
- manual configurations
  - troubleshooting 5-19
  - troubleshooting syntax errors 5-20
  - understanding error messages 5-20
  - verifying wideip.conf syntax 5-20
- Map, Network 6-1
- metrics collection
  - about TTL and timers 5-22
  - setting TTL and timer values 5-22
- monitors, file 4-1
- MX resource records 8-3

## N

- Network Map 6-1
  - and objects 6-2
  - configuring the network 6-2
  - viewing 6-2
- network traffic
  - controlling 7-1
- network, viewing 6-1
- NS resource records 8-3
- null
  - load balancing mode 5-6

**P**

- packet rate
  - load balancing mode 5-9
- password authentication 9-1
- pools 5-14
- Production Rule wizard 7-2
- production rules 7-1
  - 3dscrip 7-8
  - according to LDNS 7-16
  - according to time of day 7-14
  - actions 7-13
  - adding 7-2
  - choosing rule types 7-2
  - custom 7-8
  - defining triggers 7-5
  - deleting 7-2
  - detecting hackers 7-16
  - examples 7-14
  - executing 7-8
  - global 7-2
  - guidelines 7-8
  - inserting in wideip.conf file 7-8
  - managing 7-8
  - three basic steps 7-1
  - using Configuration utility 7-1
  - using every statement 7-12
  - using if statement 7-9
  - using scripting language 7-8
  - using when statement 7-11
  - viewing 7-2
  - wide IP 7-2
- protection from hackers
  - using production rules 7-16
- PTR resource records 8-4

**R**

- random
  - load balancing mode 5-5
- ratio
  - load balancing mode 5-4
- release notes 1-3
- resource records
  - A 8-2

**CNAME 8-3**

- MX 8-3
- NS 8-3
- PTR 8-4
- SOA 8-4
- return to DNS
  - load balancing mode 5-7
- round robin
  - load balancing mode 5-4
- round trip times
  - load balancing mode 5-9
- RSA authentication
  - generating 9-1
- rules
  - production 7-1

**S**

- scripting language
  - setting up production rules 7-8
- scripts 9-1
  - 3dns\_admin\_start 9-1
  - 3dns\_auth 9-1
  - 3dns\_dump 9-1
  - 3dns\_sync\_metrics 9-2
  - 3dns\_web\_config 9-2
  - 3dns\_web\_passwd 9-2
  - 3dnsmaint 9-2
  - 3dprint 9-3
  - 3ndc 9-4
  - big3d\_check 9-5
  - big3d\_install 9-5
  - big3d\_restart 9-6
  - big3d\_version 9-6
  - edit\_lock 9-6
  - edit\_wideip 9-6
  - install\_key 9-7
  - syncd\_checkpoint 9-7
  - syncd\_rollback 9-8
  - syncd\_start 9-9
  - syncd\_stop 9-10
- service monitors, ECV 4-1

SNMP

- client access 10-4, 10-8
- in the Configuration utility 10-9
- MIB 10-2, 10-8
- OIDs 10-6
- trap configuration 10-5

SNMP agents

- on hosts 10-11

SOA resource records 8-4

syncd\_checkpoint 9-7

syncd\_rollback 9-8

syncd\_start 9-9

syncd\_stop 9-10

syslog utility 10-7

T

technical support resources 1-3

time of day load balancing 7-14

time of day variable

- applying 7-3

timer values

- about 5-23
- and metrics collection 5-23
- configuring 5-26

To apply a combined date and time variable 7-5

Topology

- using topology records 11-1

Topology load balancing mode

- about 11-1
- configuring in pools 11-4
- configuring in wide IPs 11-2
- in a pool 11-1
- in wide IPs 11-1

topology records

- about 11-1
- in topology statements 11-1
- variables 11-6

topology statement

- variables 11-6

trap configuration 10-8

triggers 7-3

- event-based 7-5
- time-based 7-3

TTL values

- about 5-22
- and metrics collection 5-22
- configuring 5-26

U

utilities

- syslog 10-7
- viewing man pages 12-1

V

view of network 6-1

W

when statement

- guidelines 7-11
- in production rules 7-11

wide IP production rules 7-2, 7-3

wide IPs

- about 5-13
- and DNS zone file management 5-14
- configuring 5-14
- syntax 5-17
- using a last resort pool 5-18

wideip.conf file

- production rules 7-8