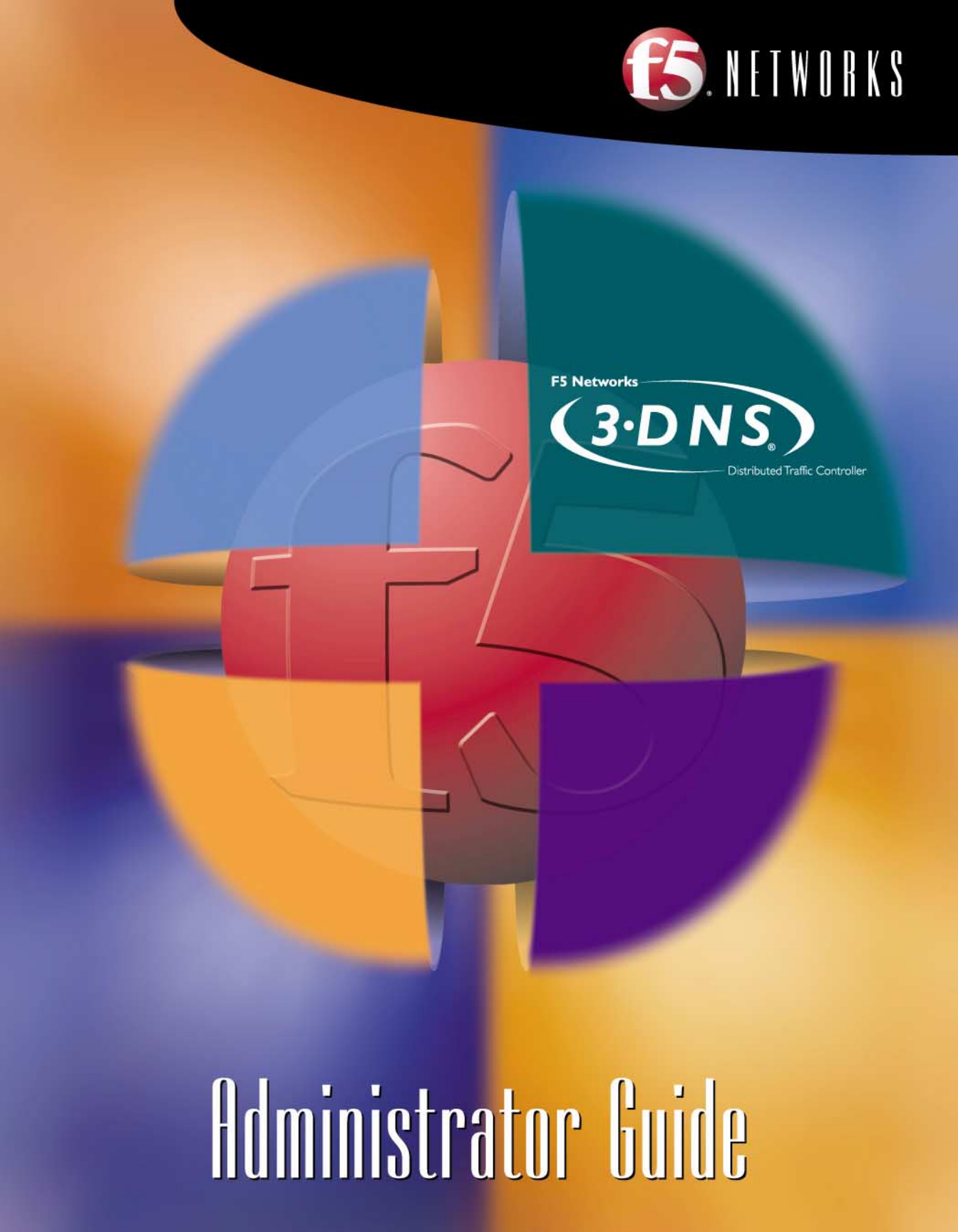

F5 NETWORKS
F5 Networks
3·DNS®
Distributed Traffic Controller
Administrator Guide

Service and Support Information

Product Version

This manual applies to version 2.1 of the 3-DNS® Distributed Traffic Controller.

Obtaining Technical Support

Web	tech.f5.com
Phone	(206) 505-0888
Fax	(206) 505-0802
Email (support issues)	support@f5.com
Email (suggestions)	feedback@f5.com

Contacting F5 Networks

Web	www.f5.com
Toll-free phone	(888) 88BIG-IP
Corporate phone	(206) 505-0800
Fax	(206) 505-0801
Email	sales@f5.com
Mailing Address	200 1st Avenue West Suite 500 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 at any time without notice.

Trademarks

F5, BIG-IP, and 3-DNS are registered trademarks of F5 Networks, Inc. SEE-IT and GLOBAL-SITE are trademarks of F5 Networks, Inc. Other product and company names are registered trademarks or trademarks of their respective holders.

Export Regulation Notice

The 3-DNS® Distributed Traffic 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® Distributed Traffic 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.

Canadian Regulatory Compliance

This class A digital apparatus complies with Canadian ICES-003.

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.

Acknowledgments

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

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

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

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

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

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

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

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

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

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

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

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

This product includes software developed by Christos Zoulas.

This product includes software developed by Charles Hannum.

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

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

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

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

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

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

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

F5 Networks Limited Warranty

This warranty will apply to any sale of goods or services or license of software (collectively, "Products") from F5 Networks, Inc. ("F5"). Any additional or different terms including terms in any purchase order or order confirmation will have no effect unless expressly agreed to in writing by F5. Any software provided to a Customer is subject to the terms of the End User License Agreement delivered with the Product.

Limited Warranty

Software. F5 warrants that for a period of 90 days from the date of shipment: (a) the media on which the software is furnished will be free of defects in materials and workmanship under normal use; and (b) the software substantially conforms to its published specifications. Except for the foregoing, the software is provided AS IS.

In no event does F5 warrant that the Software is error free, that the Product will operate with any software or hardware other than that provided by F5 or specified in the documentation, or that the Product will satisfy Purchaser's own specific requirements.

Hardware. F5 warrants that the hardware component of any Product will, for a period of one year from the date of shipment from F5, be free from defects in material and workmanship under normal use.

Remedy. Purchaser's exclusive remedy and the entire liability of F5 under this limited warranty and any other guarantee made by F5 is, at F5's option, to repair or replace any Product or component that fails during the warranty period at no cost to Purchaser. Products returned to F5 must be pre-authorized by F5 with a Return Material Authorization (RMA) number marked on the outside of the package, and sent prepaid, insured, and packaged appropriately for safe shipment. The repaired or replaced item will be shipped to Purchaser, at F5's expense, no later than 7 days after receipt by F5. Title to any returned Products or components will transfer to F5 upon receipt. F5 will replace defective media or documentation or, at its option, undertake reasonable efforts to modify the software to correct any substantial non-conformance with the specifications.

Restrictions. The foregoing limited warranties extend only to the original Purchaser, and do not apply if a Product (a) has been altered, except by F5, (b) has not been installed, operated, repaired, or maintained in accordance with F5's instructions, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (d) has been operated outside of the environmental specifications for the Product. F5's limited software warranty does not apply to software corrections or upgrades.

Support, Upgrades. F5 provides software telephone support services at no charge for 90 days following the installation of any Product: Monday through Friday, from 6 a.m. to 6 p.m. Pacific time, excluding F5's holidays. Such support will consist of responding to trouble calls as reasonably required to make the Product perform as described in the Specifications. For advisory help requests, which are calls of a more consultative nature than a standard trouble call, F5 will provide up to two hours of telephone service at no charge. Additional service for advisory help requests may be purchased at F5 Networks' then-current standard service fee. During this initial 90 day period, Customer is entitled, at no charge, to updated versions of covered software such as bug fixes, and incremental enhancements as designated by minor revision increases. In addition, Customer will receive special pricing on upgraded versions of covered Products such as new clients, new modules, and major enhancements designated by major revision increases. Customer may purchase a Maintenance Agreement for enhanced maintenance and support services.

DISCLAIMER; LIMITATION OF REMEDY: EXCEPT FOR THE WARRANTIES SPECIFICALLY DESCRIBED HEREIN, F5 DOES NOT MAKE ANY GUARANTEE OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO PRODUCTS, SPECIFICATIONS, SUPPORT, SERVICE, OR ANYTHING ELSE. F5 HAS NOT AUTHORIZED ANYONE TO MAKE ANY REPRESENTATION OR WARRANTY OTHER THAN AS PROVIDED ABOVE. F5 DISCLAIMS ANY AND ALL WARRANTIES AND GUARANTEES, EXPRESS, IMPLIED, OR OTHERWISE, ARISING WITH RESPECT TO THE PRODUCTS OR SERVICES DELIVERED HEREUNDER, INCLUDING BUT NOT LIMITED TO THE WARRANTY OF MERCHANTABILITY, THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF NON-INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. F5 WILL HAVE NO OBLIGATION OR LIABILITY, WHETHER ARISING IN CONTRACT (INCLUDING WARRANTY), TORT (INCLUDING ACTIVE, PASSIVE, OR IMPUTED NEGLIGENCE, STRICT LIABILITY, OR PRODUCT LIABILITY), OR OTHERWISE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF REVENUE, LOSS OF BUSINESS, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH ANY OF THE PRODUCTS OR OTHER GOODS OR SERVICES FURNISHED TO CUSTOMER BY F5, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

End-user Software License

IMPORTANT! READ BEFORE INSTALLING OR OPERATING THIS PRODUCT.

CAREFULLY READ THE TERMS AND CONDITIONS OF THIS LICENSE BEFORE INSTALLING OR OPERATING THIS PRODUCT: BY INSTALLING, OPERATING, OR KEEPING THIS PRODUCT FOR MORE THAN THIRTY DAYS AFTER DELIVERY, YOU INDICATE YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY CONTACT F5 NETWORKS, INC. (“F5”) TO ARRANGE FOR RETURN OF THE PRODUCT FOR A REFUND.

1. Scope. This License applies to the software for the 3-DNS® Distributed Traffic Controller, whether such software is provided separately or as an integral part of a hardware product. As used herein, the

term “Software” will refer to all such software, and the corrections, updates, new releases and new versions of such software. A product that consists of Software only will be referred to as a “Software Product” and a combination Software/Hardware product will be referred to as a “Combination Product.” All Software is licensed, not sold, by F5. This License is a legal agreement between F5 and the single entity (“Licensee”) that has acquired Software from F5 under applicable terms and conditions.

2. License Grant. Subject to the terms of this License, F5 grants to Licensee a non-exclusive, non-transferable license to use the Software in object code form solely on a single central processing unit owned or leased by Licensee. Other than as specifically described herein, no right or license is granted to Licensee to any of F5’s trademarks, copyrights, or other intellectual property rights. Licensee may make one back-up copy of any Software Product, provided the back-up copy contains the same copyright and proprietary information notices as the original Software Product. Licensee is not authorized to copy the Software contained in a Combination Product. The Software incorporates certain third party software which is used subject to licenses from the respective owners.

3. Restrictions. The Software, documentation, and the associated copyrights are owned by F5 or its licensors, and are protected by law and international treaties. Except as provided above, Licensee may not copy or reproduce the Software, and may not copy or translate the written materials without F5’s prior, written consent. Licensee may not copy, modify, reverse compile, or reverse engineer the Software, or sell, sub-license, rent, or transfer the Software or any associated documentation to any third party.

4. Export Control. F5’s standard Software incorporates cryptographic software. Licensee agrees to comply with the Export Administration Act, the Export Control Act, all regulations promulgated under such Acts, and all other laws and governmental regulations relating to the export of technical data, and equipment, and products produced therefrom, which are applicable to Licensee. In countries other than the US, Licensee agrees to comply with the local regulations regarding exporting or using cryptographic software.

5. Limited Warranty.

   a. Warranty. F5 warrants that for a period of 90 days from the date of shipment: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications. Except for the foregoing, the Software is provided AS IS. In no event does F5 warrant that the Software is error-free, that it will operate with any software or hardware other than that provided by F5 or specified in the documentation, or that the Software will satisfy Licensee’s own specific requirements.

   b. Remedy. Licensee's exclusive remedy and the entire liability of F5 under this limited warranty and any other guarantee made by F5 is, at F5's option, to repair or replace any F5 product that fails during the warranty period at no cost to Licensee. Any products returned to F5 must be pre-authorized by F5 with a Return Material Authorization (RMA) number marked on the outside of the package, and sent prepaid, insured, and packaged appropriately for safe shipment. The repaired or replaced item will be shipped to Licensee, at F5's expense, no later than 7 days after receipt by F5. Title to any returned product or components will transfer to F5 upon receipt. F5 will replace defective media or documentation or, at its option, undertake reasonable efforts to modify the Software to correct any substantial non-conformance with the specifications.

   c. Restrictions. The foregoing limited warranties extend only to the original Licensee, and do not apply if a Software Product or Combination Product (i) has been altered, except by F5, (ii) has not been installed, operated, repaired, or maintained in accordance with F5’s instructions, (iii) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident or (iv) has been

operated outside of the environmental specifications for the product. F5's limited software warranty does not apply to software corrections or upgrades.

6. Disclaimer; Limitation of Remedy. EXCEPT FOR THE WARRANTIES SPECIFICALLY DESCRIBED HEREIN, F5 DOES NOT MAKE ANY GUARANTEE OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE, SPECIFICATIONS, SUPPORT, SERVICE OR ANYTHING ELSE. F5 HAS NOT AUTHORIZED ANYONE TO MAKE ANY REPRESENTATION OR WARRANTY OTHER THAN AS PROVIDED ABOVE. F5 DISCLAIMS ANY AND ALL WARRANTIES AND GUARANTEES, EXPRESS, IMPLIED OR OTHERWISE, ARISING WITH RESPECT TO THE SOFTWARE OR SERVICES DELIVERED HEREUNDER, INCLUDING BUT NOT LIMITED TO THE WARRANTY OF MERCHANTABILITY, THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF NON-INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. F5 WILL HAVE NO OBLIGATION OR LIABILITY, WHETHER ARISING IN CONTRACT (INCLUDING WARRANTY), TORT (INCLUDING ACTIVE, PASSIVE, OR IMPUTED NEGLIGENCE, STRICT LIABILITY OR PRODUCT LIABILITY), OR OTHERWISE, FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF REVENUE, LOSS OF BUSINESS, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR OTHER GOODS OR SERVICES FURNISHED TO LICENSEE BY F5, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination. This License is effective until terminated, and will automatically terminate if Licensee fails to comply with any of its provisions. Upon termination of this License, the Licensee will destroy the Software and documentation and all copies or portions thereof.

8. Miscellaneous. This Agreement will be governed by the laws of the State of Washington, USA without regard to its choice of law rules. The provisions of the U.N. Convention for the International Sale of Goods will not apply. Any provisions found to be unenforceable will not affect the enforceability of the other provisions contained herein, but will instead be replaced with a provision as similar in meaning to the original as possible. This Agreement constitutes the entire agreement between the parties with regard to its subject matter. No modification will be binding unless in writing and signed by the parties.

F5 Networks

3·DNS®

Distributed Traffic Controller

Table of Contents

Chapter 1
Introduction to the 3-DNS Controller

Chapter 2
Preparing for Installation

Chapter 3
Setting Up the Hardware

Chapter 4
Defining the Network Setup

Chapter 5
Configuring Standard Load Balancing

Chapter 6
Configuring Specialized Load Balancing

Chapter 7
Monitoring and Administration

Chapter 8
Configuring SNMP

Chapter 9
Controlling Network Traffic Patterns

Appendix A
Wideip.conf Syntax

Appendix B
3-DNS Controller Utilities and Scripts

Appendix C
BIND 8 Information

1

Introduction to the 3-DNS Controller

- Welcome to the 3-DNS Controller
- 3-DNS Controller specifications
- Managing traffic on a global network
- What's new in version 2.1
- Finding help and technical support resources

Welcome to the 3-DNS Controller

Welcome to the *3-DNS® Controller Administrator Guide.* This guide describes how to set up the 3-DNS Controller hardware and how to set up your network and load balancing configurations, as well as other 3-DNS Controller features. The Administrator guide also includes the software specifications for the 3-DNS Controller platform, and it offers some sample configurations that can help you plan your own configuration.

3-DNS Controller specifications

The 3-DNS Controller is a network appliance that manages and balances traffic over global networks. The 3-DNS Controller manages network traffic patterns using load balancing algorithms, topology-based routing, and production rules that control and distribute traffic according to specific policies. The system is highly configurable, and its web-based and command line configuration utilities allow for easy system setup and monitoring.

The 3-DNS Controller provides a variety of features that meet special needs. For example, with this product you can:

- Guarantee multiple port availability for e-commerce sites
- Provide dynamic persistence by maintaining a connection between an LDNS IP address and a virtual server in a wide IP pool
- Restrict local clients to local servers for internationally-distributed sites
- Change the load balancing configuration according to current traffic patterns or time of day
- Customize load balancing modes
- Use network management tools to monitor 3-DNS via SNMP

Internet protocol and network management support

The 3-DNS Controller supports both standard DNS protocol and the F5 iQuery protocol (a protocol used for collecting dynamic load balancing information). The 3-DNS Controller also supports administrative protocols, such as Simple Network Management Protocol (SNMP), and Simple Mail Transfer Protocol (SMTP) (outbound only), for performance monitoring and notification of system events. For administrative purposes, you can use the F-Secure SSH client (distributed only in crypto 3-DNS Controllers) which provides a secure shell connection, **rsh**, Telnet, and FTP. The Configuration utility supports secure connections via SSL (distributed only in crypto 3-DNS Controllers), as well as standard HTTP connections.

The 3-DNS Controller's SNMP agent allows you to monitor status and current traffic flow using popular network management tools, including the Configuration utility. The SNMP agent provides detailed data such as current connections being handled for each virtual server.

Security features

The 3-DNS Controller offers a variety of security features that can help prevent hostile attacks on your site or equipment.

- **Secure administrative connections**
  crypto versions of 3-DNS Controllers support secure shell administrative connections via F-Secure SSH. The 3-DNS web server, which hosts the web-based Configuration utility, supports SSL connections as well as user authentication.
- **Secure iQuery communications**
  crypto versions of 3-DNS Controllers also support Blowfish encryption for iQuery communications between controllers running the **big3d** agent.
- **TCP wrappers**
  TCP wrappers provide an extra layer of security for network connections.

- **IP address filtering**
  The IP filtering feature, based on BSD IP packet filtering, specifically accepts or denies connections received from particular IP addresses or ranges of IP addresses.

Configuration scalability

The 3-DNS Controller is a highly scalable and versatile solution. You can configure the 3-DNS Controller to manage up to several hundred domain names, including full support of domain name aliases. The 3-DNS Controller supports a variety of media options, including Fast Ethernet, Gigabit Ethernet, and FDDI; it also supports multiple network interface cards that can provide redundant or alternate paths to the network.

Configuration and monitoring tools

The 3-DNS Controller provides the following web-based and command line administrative tools that make for easy setup and configuration.

First-Time Boot utility

The First-Time Boot utility is a wizard that walks you through the initial system set up. The utility helps you quickly define basic system settings, such as a root password and the IP addresses for the interfaces that connect the 3-DNS Controller to the network. The First-Time Boot utility also helps you configure access to the 3-DNS web server, which hosts the web-based Configuration utility, as well as the NameSurfer application that you can use for DNS zone file management.

Configuration utility

The Configuration utility is a web-based application that you use to configure and monitor the 3-DNS Controller. Using the Configuration utility, you can define the load balancing configuration along with the network setup, including data centers, sync groups, and servers used for load balancing and path probing.

In addition, you can configure advanced features such as topology settings, IP filters, and the SNMP agent. The Configuration utility also monitors network traffic, current connections, load balancing statistics, and the operating system itself.

The 3-DNS web server, which hosts the Configuration utility, provides convenient access to downloads such as the SNMP MIB and documentation for third-party applications such as NameSurfer™.

NameSurfer application

The NameSurfer application is a third-party application, produced by Data Fellows, that automatically configures DNS zone files associated with domains handled by the 3-DNS Controller. You can use NameSurfer to configure and maintain additional DNS zone files on 3-DNS Controllers that run as master DNS servers. The Configuration utility provides direct access to the NameSurfer application, as well as the corresponding documentation for the application.

3-DNS Maintenance menu

The 3-DNS Maintenance menu is a command line utility that executes scripts which assist you in configuration and administrative tasks, such as installing the latest version of the **big3d agent** on all your systems, or editing the load balancing configuration files. You can use the 3-DNS Maintenance menu directly on the 3-DNS Controller, or you can use the menu when connected to the controller via a remote shell, such as the SSH client (on crypto 3-DNS Controllers only), or a standard **rsh** client (if **rsh** is configured).

Browser support

The Configuration utility, which provides web-based access to the 3-DNS Controller system configuration and features, supports the following browser versions:

- Netscape Navigator 4.5 or later

- Microsoft Internet Explorer, version 4.02 or later

System synchronization options

The 3-DNS Controller sync group feature allows you to automatically synchronize configurations from one 3-DNS Controller to the other 3-DNS Controllers in the network, simplifying administrative management. The synchronization feature offers a high degree of administrative control. For example, you can set the controller to synchronize a specific configuration file set, and you can also set which 3-DNS Controllers in the network receive the synchronized information and which ones do not.

Configurable data collection for server status and network path data

The 3-DNS Controller platform includes a **big3d** agent, which is an integral part of 3-DNS Controller load balancing. The **big3d** agent continually monitors the availability of the servers that the 3-DNS Controller load balances. It also monitors the integrity of the network paths between the servers that host the domain and the various client LDNS servers that attempt to connect to the domain. The **big3d** agent runs on 3-DNS Controllers and BIG-IP Controllers distributed in various locations in your network. Each **big3d** agent broadcasts its collected data to all of the 3-DNS Controllers in your network, ensuring that all 3-DNS Controllers work with the latest information.

The **big3d** agent offers a variety of configuration options which allow you to choose the types of data collection methods you want to use. For example, you can configure the **big3d** agent to track the number of hops (intermediate system transitions) along a given network path, and you can also set the **big3d** agent to collect host server performance information using the SNMP protocol.

Redundant system configurations

A ***redundant system*** is essentially a pair of 3-DNS Controller units, one operating as an active unit responding to DNS queries, and one operating as a standby unit. If the active unit fails, the standby unit takes over and begins to respond to DNS queries while the other controller reboots and becomes a standby unit.

The 3-DNS Controller actually supports two methods of checking status of the peer system:

- **Hardware-based fail-over**
  In a system that has been set up with hardware-based fail-over, the two units in the system are connected to each other directly using a fail-over cable attached to the serial ports. The standby controller checks on the status of the active controller every second using this serial link. The controllers check on each other's status using that link.
- **Network-based fail-over**
  In a system that has been set up with network-based fail-over, the two units in the system communicate with each other across an Ethernet network instead of going across a dedicated fail-over serial cable. The standby controller checks on the status of the active controller every second using the Ethernet. The controllers check each other's status using that link.

Note

In a network-based fail-over configuration, the standby 3-DNS Controller immediately takes over if the active unit fails. If a client had queried the failed controller, and not received an answer, it automatically re-issues the request (after 5 seconds) and the standby unit, functioning as the active controller, responds.

IP packet filtering

The 3-DNS Controller supports easy configuration of the BSD operating system method of IP packet filtering. In the Configuration utility, you can configure individual IP packet filters, which can control both in-bound and out-bound network traffic.

For example, you can specify a single IP address or a range of IP addresses from which the 3-DNS Controller either accepts or denies network traffic. You can also specify one or more IP addresses to which you specifically want to allow or prevent out-bound connections.

Managing traffic on a global network

This section provides a brief overview of how 3-DNS Controllers work within a global network and how they interact with BIG-IP Controllers and host machines in the network. The section also illustrates how the 3-DNS Controller works with the **big3d** agents that run in various locations in the network, as well as the LDNS servers that make DNS requests on behalf of clients connecting to the Internet.

The following sample configuration shows 3-DNS Controllers that load balance connections for a sample Internet domain named **domain.com.**

A sample network layout

3-DNS Controllers sit in specific data centers in your network, and work in conjunction with BIG-IP Controllers and with generic host servers that also sit in your network data centers. All 3-DNS Controllers in the network can receive and respond to DNS resolution requests from the LDNS servers that clients use to connect to the domain.

Figure 1.1 illustrates the layout of the 3-DNS Controllers, BIG-IP Controllers, and host servers in the three data centers. The Los Angeles data center houses one 3-DNS Controller and one BIG-IP Controller, as does the New York data center. The Tokyo data center houses only one 3-DNS Controller and one host server.

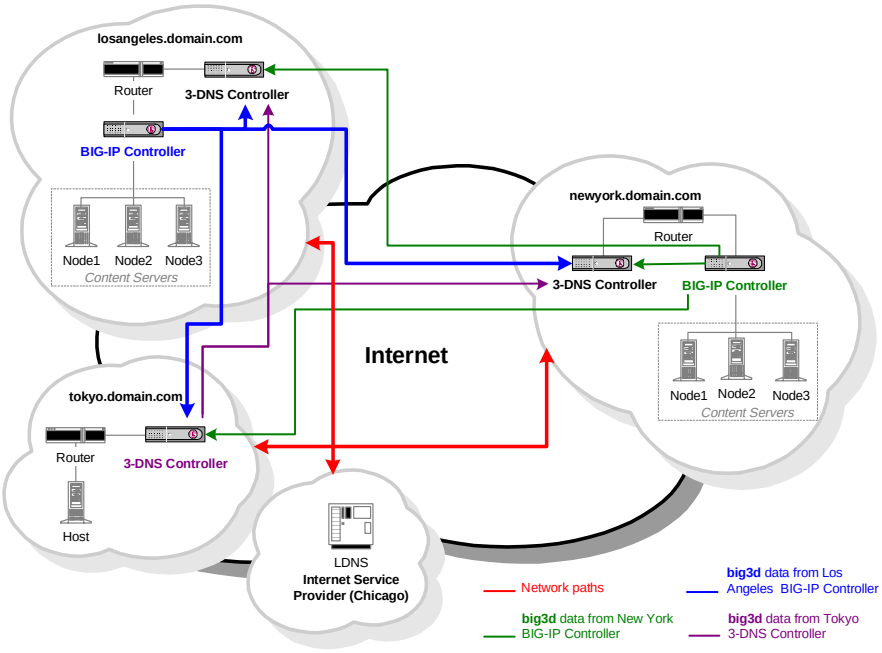


Figure 1.1 *A sample network layout*

In the Los Angeles and New York data centers, the **big3d** agent runs on the BIG-IP Controller, but in the Tokyo data center, the **big3d** agent runs on the 3-DNS Controller. Each **big3d** agent collects information about the network path between the data center where it is running and the client's LDNS server in Chicago, as illustrated by the red lines. Each **big3d** agent also broadcasts the network path information it collects to the 3-DNS Controllers running in each data center, as illustrated by the green, blue, and purple lines.

◆ Note

*Each BIG-IP Controller and 3-DNS Controller in a data center typically runs a **big3d** agent.*

Synchronizing configuration information and broadcasting performance metrics

3-DNS Controllers typically work in sync groups where a group of controllers shares load balancing configuration settings. In a sync group, any controller that has new configuration changes can broadcast the changes to any other controller in the sync group, allowing for easy administrative maintenance. To distribute metrics data among the controllers in a sync group, the ***principal 3-DNS Controller*** sends requests to the **big3d** agents in the network, asking them to collect specific performance and path data. Once the **big3d** agents collect the data, they each broadcast the collected data to all controllers in the network, again allowing for simple and reliable metrics distribution.

Using a 3-DNS Controller as a standard DNS server

When a client requests a DNS resolution for a domain name, DNS sends the request to the 3-DNS Controller that is authoritative for the zone (running as a master DNS server for the domain). The 3-DNS Controller chooses the best available virtual server out of a pool, and then returns a standard DNS answer record (an **A** record) to the requesting LDNS server. The LDNS server uses the answer for the period of time defined within the **A** record. Once the answer expires, however, the LDNS server must request name resolution all over again to get a fresh answer.

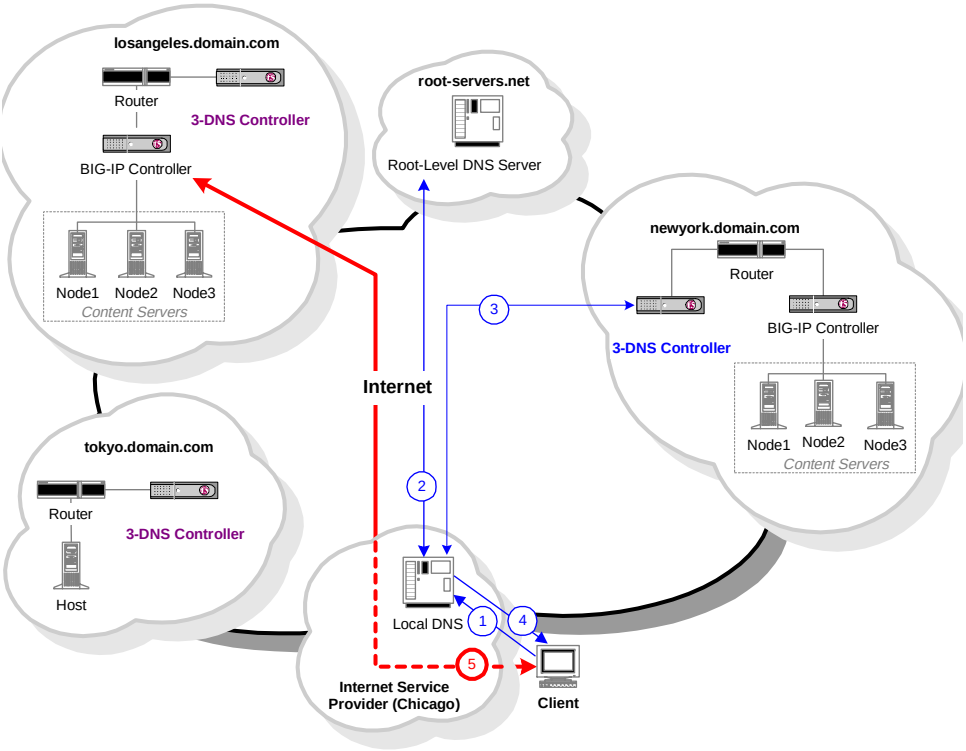


Figure 1.2 *Name resolution process*

Figure 1.2 illustrates the specific steps in the name resolution process.

1. The client connects to an Internet Service Provider (ISP) and queries the LDNS to resolve the domain name **www.domain.com**.
2. If the information is not already in the LDNS server's cache, the LDNS server queries a root server (such as InterNIC's root servers). The root server returns the IP address of a DNS associated with **www.domain.com**, which in this case runs on the 3-DNS Controller.

3. The LDNS then connects to the 3-DNS Controller looking to resolve the **www.domain.com** name. The 3-DNS Controller uses a load balancing mode to choose an appropriate server to receive the connection, and returns the server's IP address to the LDNS.
4. The LDNS ends the connection to the 3-DNS Controller and passes the IP address to the client.
5. The client connects to the IP address via the ISP.

Note

The dotted portion of line 5 indicates that the actual hardware for this step is not shown, due to the number of ways ISPs can configure their networks. The actual machines that handle all other transaction events are shown, so all other lines are solid.

Load balancing connections across the network

Each of the 3-DNS Controller load balancing modes can provide efficient load balancing for any network configuration. The 3-DNS Controller bases load balancing on pools of virtual servers. When a client requests a DNS resolution, the 3-DNS Controller uses the specified load balancing mode to choose a virtual server from a pool of virtual servers. The resulting answer to this resolution request is returned as a standard **A** record.

Although some load balancing configurations can get complex, most load balancing configurations are relatively simple, whether you use a basic, static load balancing mode or an advanced, dynamic load balancing mode. More advanced configurations can incorporate multiple pools, as well as advanced traffic control features, such as topology or production rules. (For a list of individual load balancing modes, see *Choosing a load balancing mode*, on page 2-27.

Working with BIG-IP Controllers and other products

The 3-DNS Controller balances connections across a group of virtual servers that run in different data centers throughout the network. You can manage virtual servers from the following types of products:

- **BIG-IP Controllers**
  A BIG-IP Controller virtual server maps to a series of content servers.
- **Generic hosts**
  A host virtual server can be an IP address or an IP alias that hosts the content.
- **Other load balancing products**
  Other load balancing products map virtual servers to a series of content hosts.

Figure 1.3 illustrates the hierarchy of virtual server management in our sample configuration.

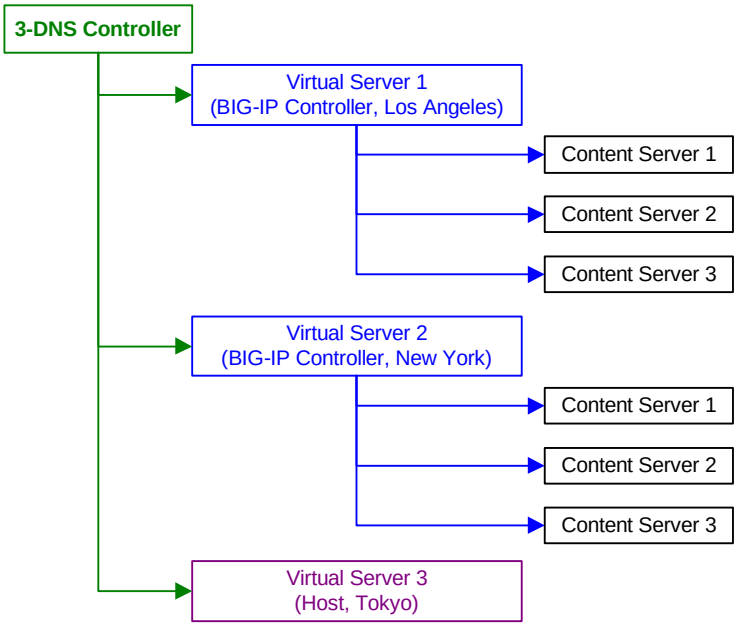


Figure 1.3 *Load balancing management*

How 3-DNS Controllers differ from BIG-IP Controllers

While both controllers provide load balancing, one of the significant differences between the 3-DNS Controller and the BIG-IP Controller is that the 3-DNS Controller responds to DNS requests issued by an LDNS on behalf of a client, while the BIG-IP Controller provides connection management between the client and the back-end server.

Once the 3-DNS Controller returns a DNS answer to an LDNS, the conversation between the LDNS and the 3-DNS Controller ends, and the client connects to the IP address returned by the 3-DNS Controller. Unlike 3-DNS, the BIG-IP Controller sits between the client and the content servers. It manages the client's entire conversation with the content server.

What's new in version 2.1

The 3-DNS Controller offers the following new features in version 2.1.

Dynamic Persistence

The 3-DNS Controller now provides dynamic persistence, enabling you to maintain a connection between an LDNS and a particular virtual server in a wide IP, rather than load balancing the connection to any available virtual server. For information on how to configure this option, view the Configuration utility online help for the Edit Wide IP screen.

Advanced load balancing to the Cisco LocalDirector

The 3-DNS Controller can now acquire metrics from the Cisco LocalDirector and load balance to a LocalDirector using both basic and advanced load balancing modes. For more information on this feature, see *Configuring host SNMP settings*, on page 4-15

New load balancing mode

The 3-DNS Controller has a new, advanced load balancing mode called VS Capacity. This dynamic load balancing mode is a stand-alone mode as well as part of the Quality of Service (QOS) mode. For more information on this load balancing mode, see *VS Capacity mode*, on page 5-10.

iQuery enhancements

The TCP protocol has been added, so you can now choose UDP or TCP as an iQuery transport option when defining BIG-IP and 3-DNS Controllers. For information on how to configure this option, view the Configuration utility online help for the Add New 3-DNS Controller and Add New BIG-IP Controller screens.

Also, the iQuery protocol is now backward compatible with 3-DNS Controller, version 2.0.X.

BIND Upgrade

The 3-DNS Controller, version 2.1 incorporates BIND version 8.2.2 p5.

Configurable probe protocols

You can now specify precisely which protocols to use when probing LDNS servers and hosts, and in what order to use the protocols. In addition, we have added two new protocols to the list, DNS_VER and DNS_DOT. For information on how to configure this option, view the Configuration utility online help for the System - Metric Collection screen.

Enable/disable option to change status of objects

You can now use the Configuration utility to change the status of objects, and either disable or enable the objects. The objects you can change the status of include wide IPs, wide IP pools, sync groups, data centers, 3-DNS Controllers, BIG-IP Controllers, host servers, and virtual servers. There is a hierarchy among these objects. If one object is disabled, all objects that the object owns are implicitly disabled. For example, by disabling a data center, you implicitly disable all of its servers and virtual servers. To enable these servers and virtual servers, you must first enable the data center.

You can also view the status of the various objects in each object's statistics screen. For information on how to configure this option, view the Configuration utility online help for the specific object for which you want to change status.

Scripts to back up and restore 3-DNS Controller configurations

The new **3dns_backup** script creates a backup file that, once restored, configures a 3-DNS Controller with the same configuration as the 3-DNS Controller that created the backup. You can copy the backup file to another computer system or to a diskette.

The **3dns_restore** script restores a backup file that was created using the **3dns_backup** script, and configures a 3-DNS Controller with the same configuration as the 3-DNS Controller that originally created the backup.

Multiple pool support using the Configuration utility

You can now configure multiple pools for specific wide IPs using the Configuration utility. These pools may now contain both host server and BIG-IP Controller virtual servers.

3-DNS Controller subnetting

When you use NameSurfer, subnetting management is now integrated into the Configuration utility.

- When you create a wide IP, the NameSurfer zone files allow forward and reverse references to the virtual servers and the wide IP itself.
- When you delete a virtual server or wide IP using the Configuration utility, the 3-DNS Controller now deletes the appropriate forward and reverse records from the NameSurfer zones.
- When you add virtual servers to the wide IP, the 3-DNS Controller now creates the appropriate forward and reverse records in the NameSurfer zones.
- When you add or change a wide IP alias, the 3-DNS Controller now makes the appropriate changes to the NameSurfer zones.

New probing exclusion lists

You can now create a probing exclusion list, that contains a group of LDNS IP addresses whose paths the 3-DNS Controller will not probe. There are three different types of ACLs:

- **probe_acl**
  The 3-DNS Controller restricts any **big3d** agent from probing this group of LDNS servers. For example, in the **wideip.conf** file you would type:

```
probe_acl {
    10.20.30.0/24
    192.168.0.0/16
    209.221.0.0/16
}
```

- **hops_acl**
  The 3-DNS Controller restricts any **big3d** agent from tracerouting this group of LDNS servers. For example, in the **wideip.conf** file you would type:

```
hops_acl {
    10.20.30.0/24
    192.168.0.0/16
    209.221.0.0/16
}
```

- **discovery_acl**
  The 3-DNS Controller restricts any **big3d** agents from performing port discovery on this group of LDNS servers. For example, in the **wideip.conf** file you would type:

```
discovery_acl {
    10.20.30.0/24
    192.168.0.0/16
    209.221.0.0/16
```

Rollup and rollback scripts

When upgrading the product, the **upgrade_install** script rolls up the old installation. This enables you, if necessary, to uninstall the newest installation and restore the old one.

Network time protocol (NTP) support

When installing the 3-DNS Controller, you now have the option to synchronize to a public time server. For more information on this feature, see*Configuring NTP clocks*, on page 3-15.

New variable to check principal/receiver status

The 3-DNS Controller includes a new **timer_sync_state** variable which enables you to specify 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 returns to a principal state, and the temporary principal returns to a receiver state.

For information on how to configure this option, view the Configuration utility online help for the System - Timers & Task Intervals screen.

Improved log messages

The 3-DNS Controller now supplies more detailed log messages.

Single interface management

You can now manage F5 products from a single screen. For example, the BIG-IP Controllers screen shows all BIG-IP Controllers managed by the 3-DNS Controller. If you click a BIG-IP Controller from the Launch column, the Configuration utility for the corresponding BIG-IP Controller opens.

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 a PDF version of this administrator guide. Note that some 3-DNS Controller upgrades replace the online administrator guide with an updated version of the guide.
  - The Configuration utility also has online help for each screen. Simply 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 the command followed by the question mark option (**-?**), 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 all third-party software included with the 3-DNS Controller, including NameSurfer and GateD.
- **Technical support via 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 administrator guides (in PDF and HTML formats). To access this site you must first email **askf5@f5.com** and obtain a customer ID and a password.

F5 Networks
3·DNS®
Distributed Traffic Controller

2

Preparing for Installation

- Reviewing the installation tasks
- Planning issues for the hardware setup
- Planning issues for the network setup
- Planning issues for the load balancing configuration
- Using advanced traffic control features
- Planning DNS zone file management

Reviewing the installation tasks

This chapter covers the planning issues that you need to address before you install and set up the 3-DNS Controller. It explains the three different phases of an installation, along with load balancing configuration options and important DNS planning issues. If you plan to do a simple configuration, briefly review the hardware setup and network setup requirements, along with the load balancing options. If you plan to do a more advanced configuration, we recommend that you read about each advanced feature you want to work with in Chapter 6 before you begin the installation.

Be sure to review the DNS zone file management section included later in this chapter. It provides an important overview of the process involved in setting up a 3-DNS Controller that runs as a master DNS server, as well as integrating 3-DNS Controllers into networks that have an existing master DNS server.

Understanding the installation phases

You can set up a basic 3-DNS Controller configuration, or an advanced 3-DNS Controller configuration. Regardless of whether you are creating a basic or an advanced configuration, you always go through three installation phases:

- **Hardware setup**
  The hardware setup phase simply gets the hardware up and running, and connected to the network. This includes completing the configuration settings that allow remote access for command line access via a UNIX shell, and for the 3-DNS web server which hosts the Configuration utility. You must do the hardware setup on all 3-DNS Controllers in your network.
- **Network setup**
  During the network setup phase, you define the network layout of the servers that the 3-DNS Controller uses for load balancing and path probing. You define the different data centers in the network, as well as the equipment that runs in each data center. You also set up a ***sync group***, which is a group of 3-DNS Controllers that share system configurations. Note that at this point, you can define the configuration on the 3-DNS Controller

that is the principal controller in the sync group, and then all other controllers in the sync group automatically copy the configuration from the principal controller.

- **Load balancing configuration**
  The load balancing configuration is the last phase of a 3-DNS Controller installation. This phase can be simple and fast if you are setting up a very basic configuration, particularly if you allow the NameSurfer application to handle the DNS zone file management for you. If you are planning to use a more advanced load balancing configuration, however, you may need to do some additional planning.

DNS zone file integration happens at different times, depending on how you decide to handle DNS zone file management, and depending on whether you are installing new 3-DNS Controllers in your network, or upgrading existing 3-DNS Controllers in your network. Issues relating to DNS zone file management are covered in *Planning DNS zone file management*, on page 2-32.

Working with configuration tools

When you configure the 3-DNS Controller, you have two configuration tool options:

- We recommend that you use the browser-based Configuration utility in conjunction with the NameSurfer application. Using these tools together provides two significant benefits that make for simplified system administration. First, when you make configuration changes to any 3-DNS Controller in the network, the sync feature automatically broadcasts the changes to all other 3-DNS Controllers in the network. Second, when you add or modify wide IP definitions in the Configuration utility, the NameSurfer application automatically makes the appropriate updates to the DNS zone files. You do not need to do any DNS zone file maintenance yourself.
- If you prefer, you can manually configure the 3-DNS Controller using the **Edit 3-DNS Configuration** command on the 3-DNS Maintenance menu (a command line tool). However, this method requires that you make corresponding changes to DNS

zone files manually, which can be quite complex. It also requires that you individually configure each 3-DNS Controller in the network.

If you have worked with previous versions of the 3-DNS Controller, you can view statement syntax in the Configuration utility. To display the existing configuration at any time, click the View Configuration button in the Conf column of certain screens in the configuration utility.

WARNING

We recommend that you do not switch between using the Configuration utility and manually configuring the 3-DNS Controller. If you manually configure a 3-DNS Controller and then attempt to use the Configuration utility to modify the configuration, you risk losing custom configuration settings not supported by the Configuration utility.

Planning issues for the hardware setup

In the hardware setup phase, you connect the 3-DNS Controller hardware to the network, and run the First-Time Boot utility on each of the 3-DNS Controllers in the network. The First-Time Boot utility is a command line wizard that helps you define basic required system settings such as the IP address, host name, root user ID, and other information necessary for the controller and its web server to be accessible over the network.

When you run the First-Time Boot utility, you must work on a keyboard and monitor, or a serial terminal, connected directly to the 3-DNS Controller. However, after you complete the hardware setup phase, you can finish the remaining installation tasks from a remote administrative workstation.

Gathering basic information

The prompts in the First-Time Boot utility are generally self-explanatory, and you should be able to answer them as long as you prepare the following information before you start running the utility:

- Root administration password for each controller
- IP addresses for each network interface, and a shared IP alias for each redundant system
- Host names used on each network interface
- IP addresses of remote workstations that need to access the controllers
- Time zones for each controller
- IP addresses and remote connection requirements (**ssh** versus **rsh**) for all 3-DNS Controllers and BIG-IP Controllers in the network configuration
- Authentication certificate settings (company name, address, etc.) for 3-DNS web servers that run on crypto 3-DNS Controllers

Addressing special hardware configuration issues

Before you start the hardware setup, you may want to review the following items which address configuration and management issues for redundant systems, systems that use more than one network interface, and DNS zone file management.

Are you setting up a stand-alone unit or a redundant system?

If you are setting up a stand-alone unit, you need one IP address and host name for each of the interfaces you plan to connect to the network. If you are setting up a redundant system, you need one IP address for each network interface card in each unit, as well as a shared IP alias for the primary network interface, and a shared IP alias for the secondary network interface (if you are connecting the redundant system to more than one network).

If you are setting up a redundant system, are you using hardware-based fail-over or network-based fail-over?

Hardware-based fail-over is a redundant system that connects two 3-DNS Controller units directly to each other using a fail-over serial cable. ***Network-based fail-over*** is a redundant system where two units are either connected to each other directly using an Ethernet cable, or they are connected indirectly via an Ethernet network. Of the two units in a redundant system, one runs as the ***active unit***, managing all DNS resolution requests, and the other runs as the ***standby unit***, waiting to take over in case the active unit fails and reboots. The communication between the units, such as fail-over notification, runs across either the fail-over cable in the hardware-base redundant system, or the network in the network-based redundant system.

When you run the First-Time Boot utility, it prompts you to enter the IP address of the other unit in the system.

What events trigger a fail-over in a redundant system?

The 3-DNS Controller tracks two key aspects of the system to validate system performance. In a redundant system, there are two events that indicate a system failure and trigger a fail-over.

- If the **named** daemon becomes unresponsive, or if you manually stop the daemon using the **ndc stop** or **ndc restart** commands, the 3-DNS Controller treats this as system failure and initiates a fail-over.
- If the 3-DNS Controller fails to detect any traffic on its network interfaces, it attempts to create traffic to test the integrity of the interface. If the test fails, the 3-DNS Controller treats this as a system failure and initiates a fail-over.

How do redundant systems work with the sync group feature?

If you include a redundant system in a sync group, you include the system by specifying the system's shared IP alias. When a controller in the sync group broadcasts new information to the redundant system, only the active unit receives the updated information. The moment a fail-over occurs and the standby unit becomes active, the next synchronization check compares

timestamps on the configuration files and immediately updates the unit with the current system configuration, path metrics, and DNS zone files.

If you have a sync group that includes a redundant system, you may want to make test changes to the configuration on the standby unit, without having those changes synchronized to the other controllers in the group. As long as the unit runs as a standby, the changes remain local. Once you validate your changes and you are ready to broadcast them to the other controllers in the group, you can run the **bigpipe fo** command on the active unit to force a fail-over. After the system fails over to the standby unit with your new configuration changes, those changes broadcast to the other controllers in the group during the next synchronization pass.

Note that you always run the risk of having the active unit initiate a fail-over to the standby unit before you can verify that the configuration changes are complete. Although the 3-DNS Controller does not commit any configuration changes that violate syntax rules, there is always the possibility that a fail-over may occur before you complete your configuration changes.

Are you planning on using more than one network interface?

The First-Time Boot utility prompts you to configure the primary network interface, and then asks if you want to configure more interfaces, or if you want to skip to the next section of the utility. If you want to configure another network interface, you simply enter the same type of information you entered for the first interface. The other interfaces can connect to a separate network, or they can act as redundant paths to the same network that the first interface is connected to.

Do you want to set up automatic DNS zone file management?

The First-Time Boot utility asks you if you want to use the NameSurfer application as the master for DNS zone files. We recommend that you always run NameSurfer as the master for DNS zone files. When you define or modify wide IPs in the Configuration utility, NameSurfer automatically makes the

corresponding changes to the DNS zone files. The NameSurfer application also provides you with easy management of high-level domain zone files unrelated to the wide IP configuration.

If you plan on transferring existing BIND files from a master DNS server to the 3-DNS Controller, you do not configure NameSurfer when you run the First-Time Boot utility; you configure the application later on in the installation process. For more details about this and other DNS zone file management issues, see *Planning DNS zone file management*, on page 2-32.

Planning issues for the network setup

After you finish running the First-Time Boot utility and get each controller connected to the network, you can do the network setup and load balancing configuration on one controller, and let the sync group feature automatically broadcast the configuration to the other controllers in the network. You do not have to configure 3-DNS Controllers individually, unless you are planning an advanced configuration that requires different configurations for different data centers, or you are doing the configuration manually.

During the network setup phase, you define three basic aspects of the network layout, in the following order:

- **Data centers**
  Data centers are the physical locations that house the equipment you use for load balancing.
- **Servers**
  The servers you define in the network setup include only the 3-DNS Controllers, BIG-IP Controllers, and host machines that you use for load balancing.

- **Sync group**
  A ***sync group*** defines the group of 3-DNS Controllers that shares configuration settings and path data.

Note

During the network setup phase of configuration, we recommend that you connect to the 3-DNS Controller from a remote workstation where you can complete the remaining configuration tasks using the web-based Configuration utility.

Defining data centers and servers

It is important that you define all of your data centers before you begin defining servers. When you define a server, you specify the data center where the server runs by choosing a data center from the list of data centers you have already defined.

To define a data center, you need only specify the data center name. To define a server, however, you need to specify the following items:

- Server type (3-DNS Controller, BIG-IP Controller, or host)
- Server IP address (or shared IP alias for redundant systems)
- Name of the data center where the server runs
- **big3d** agent settings (BIG-IP Controllers and 3-DNS Controllers only)
- Virtual servers managed by the server (BIG-IP Controllers and hosts only)
- SNMP host probing settings (hosts only)

The most important part of planning data centers and servers is to decide how to set up the **big3d** agent, and which ports you need to open for communications between the controllers in your network. The following sections in this chapter, *Setting up data collection with the big3d agent*, on page 2-9, and *Setting up communications between 3-DNS Controllers, BIG-IP Controllers, and big3d agents*, on page 2-17, provide help with determining how both of these issues affect your installation.

Understanding how the time tolerance variable affects sync groups

The time tolerance value is a global variable that defines 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 greater than the time tolerance, 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 was running only 2 seconds ahead of the other, the time settings would remain unchanged. The values are 0, 5, and higher (values of 1-4 are automatically set to 5, and 0 turns off time syncing). The default setting is **10** seconds.

The time setting on 3-DNS Controllers is important because a 3-DNS Controller compares time stamps on files when deciding whether to synchronize files with other 3-DNS Controllers in the sync group.

Setting up data collection with the big3d agent

The **big3d** agent collects performance information on behalf of the 3-DNS Controller. The **big3d** agent runs on both 3-DNS Controllers and BIG-IP Controllers. The default setting is to run a **big3d** agent on all controllers in the network, but you can turn off the **big3d** agent on any controller at any time.

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

- **Installing big3d agents on BIG-IP Controllers**
  Each time you receive new 3-DNS Controller software, the software includes the latest version of the **big3d** agent. You need to distribute that copy of the **big3d** agent to the BIG-IP Controllers in the network. Use the **3dnsmaint** menu item to automatically install the appropriate version of the **big3d** agent for each version of the BIG-IP Controller. See the release notes

provided with the 3-DNS Controller software for information about which BIG-IP Controller versions the current **big3d** agent supports.

- **Specifying which factories a specific big3d agent manages**
  When you define BIG-IP Controller 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 BIG-IP Controllers and 3-DNS Controllers in the network. These planning issues are discussed in *Setting up communications between 3-DNS Controllers, BIG-IP Controllers, and big3d agents*, on page 2-17.

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
- **BIG-IP virtual server performance**
  The **big3d** agent calculates the number of connections to a virtual server defined on a BIG-IP Controller. The number of connections is used to determine the best virtual server when using the Least Connections load balancing mode.

Installing the big3d agent on BIG-IP Controllers

To install the **big3d** agent on the BIG-IP Controllers in your network, log on to the 3-DNS Controller using either a remote shell, or the serial terminal or keyboard and monitor attached directly to the controller. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu and choose the I**nstall and Start big3d** command.

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_VER protocols. This factory queries host virtual servers and LDNSs. Host virtual servers are checked to determine their **up** or **down** state. For LDNSs, the probing factory uses the response 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, and 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 and 3-DNS Controller in the network runs a **big3d** agent using five prober factories, one SNMP factory, one discovery factory, no hops factories, and the two default 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 2.1 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
Needs Probe	The **big3d** agent has never collected data for the LDNS, or the data has expired.
Idle	The **big3d** agent successfully collected data for the LDNS, and is waiting for the next collection request.
In Probe	The **big3d** agent is currently collecting data for the LDNS.
Needs Discovery	The **big3d** 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.
In Discovery	The **big3d** agent is currently running the LDNS through a discovery factory to look for an alternate available port.
Suspended	The **big3d** agent failed to discover a port open for data collection, and the LDNS is no longer eligible for data collection requests.

Table 2.1 *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 and 3-DNS Controller. 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 time-to-live (TTL**)** settings that you assign to the different types of data the agents collect, and the number of factories that each agent runs. The shorter the TTLs, the more frequently the agent needs to refresh the data. While short TTLs 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 TTL settings for round trip time data so that it refreshes more often, but set high TTL settings for hops data because it does not need to be refreshed often.

Setting up SNMP probing for hosts

The host probing feature uses SNMP conversations to collect performance data for the host. The 3-DNS Controller uses this performance data for advanced load balancing modes such as Packet Rate and Quality of Service. The **big3d** agent uses a special SNMP factory to collect metrics from host servers. The SNMP factory establishes a conversation with an SNMP agent running on a given host server. From the SNMP conversation, the factory determines the following types of information about the host:

- Memory utilization
- CPU utilization
- Disk space utilization
- Packet rate

The Configuration utility displays the host metrics in the Host statistics screen. The 3-DNS Controller bases the advanced load balancing decisions on the packet rate metrics, but the Host screen displays the other metrics as well, for your convenience.

Configuration issues

The SNMP probing feature requires that each host run an SNMP agent, and that there is open network communication between the hosts and the **big3d** agents in the data centers. Certain firewall configurations block SNMP communications, and you may need to verify that the firewalls in your network allow SNMP traffic to pass through. The 3-DNS Controller supports the following common SNMP agents for host probing:

- **Generic**
  A generic SNMP agent is an SNMP agent that provides OIDS as specified in the RFC1213 document.
- **UCD SNMPD**
  This free SNMP agent is provided by the University of California at Davis. It is available on the web at **http://ucd-snmp.ucdavis.edu,** or you can download the **ucd-snmp.tar.gz** file from **ftp://ucd-snmp.ucdavis.edu**.
- **Solstice Enterprise**
  This SNMP agent is a product of Sun Microsystems.
- **Windows NT 4.0 SNMP**
  This SNMP matrix agent is a product of Microsoft and is distributed with the Microsoft Windows NT 4.0 server.
- **Cisco LDV2**
  This SNMP agent is a product of Cisco and is distributed with the Cisco LocalDirector, version 2.X.
- **Cisco LDV3**
  This SNMP agent is a product of Cisco and is distributed with the Cisco LocalDirector, version 3.X.

In addition to properly configuring these agents on the hosts themselves, you need to specify SNMP host probing settings in two places in the 3-DNS Controller configuration. First, when you define a BIG-IP Controller or 3-DNS Controller server, you set the **big3d** agent to run at least one SNMP factory. Second, when you define the host servers, you configure specific SNMP agent settings for each host. For example, you need to specify the type of agent running on the host as well as the community string that allows access to the SNMP agent.

Planning sync groups

A ***sync group*** is a group of 3-DNS Controllers that share information. In a sync group, a ***principal*** 3-DNS Controller issues requests to the **big3d** agents to gather metrics data. Both the principal controller and the ***receiver*** 3-DNS Controllers in the group receive broadcasts of metrics data from the **big3d** agents. All controllers in the group also receive broadcasts of updated configuration settings from the 3-DNS Controller that has the latest configuration changes.

When you define the sync group, select 3-DNS Controllers from the list of servers you have already defined. The sync group lists the controllers in the order in which you selected them. The first controller in the list is the principal 3-DNS Controller. The remaining controllers in the list are receiver 3-DNS Controllers. If the principal controller becomes disabled, the next controller in the list becomes the principal 3-DNS Controller until the original principal controller comes back online.

Understanding how sync groups work

The sync group feature synchronizes individual configuration files, such as **wideip.conf** and other files that store system settings. You have the option of adding files to the synchronization list.

The controllers in a sync group operate as peer servers. At set intervals, the **sync** daemon compares the timestamps of the configuration files earmarked for synchronization on all of the controllers. If the timestamp on a specific file differs between controllers, the controller with the latest file broadcasts the file to all of the other controllers in the group.

Setting up communications between 3-DNS Controllers, BIG-IP Controllers, and big3d agents

There are three different communication issues that you need to resolve when you set up communication between controllers running in your network:

- **3-DNS Controllers communicating with other 3-DNS Controllers**
  To allow 3-DNS Controllers to communicate with each other, you must set up **ssh** and **scp** tools for crypto controllers (those which use **ssh** and **scp**) that communicate with other crypto controllers, and you must set up **rsh** and **rcp** tools for controllers that communicate with non-crypto controllers (those which do not use **ssh** and **scp**).
- **3-DNS Controllers communicating with BIG-IP Controllers**
  To allow 3-DNS Controllers to communicate with BIG-IP Controllers, you address the same **ssh** and **rsh** issues. Crypto controllers communicating with other crypto controllers can use **ssh** and **scp**, but controllers communicating with non-crypto controllers require **rsh** and **rcp**.
- **3-DNS Controllers communicating with big3d agents**
  To allow communications between **big3d** agents and the 3-DNS Controller, you need to configure iQuery ports on both the 3-DNS Controllers and the BIG-IP Controllers that run the **big3d** agent.

Note

*Enabling **rsh** and **rcp** does not prevent crypto 3-DNS Controllers from using encryption when they communicate with other crypto 3-DNS Controllers.*

Figure 2.1 shows the ports necessary for administrative communication between individual 3-DNS Controllers, and also between 3-DNS Controllers and administrative workstations.

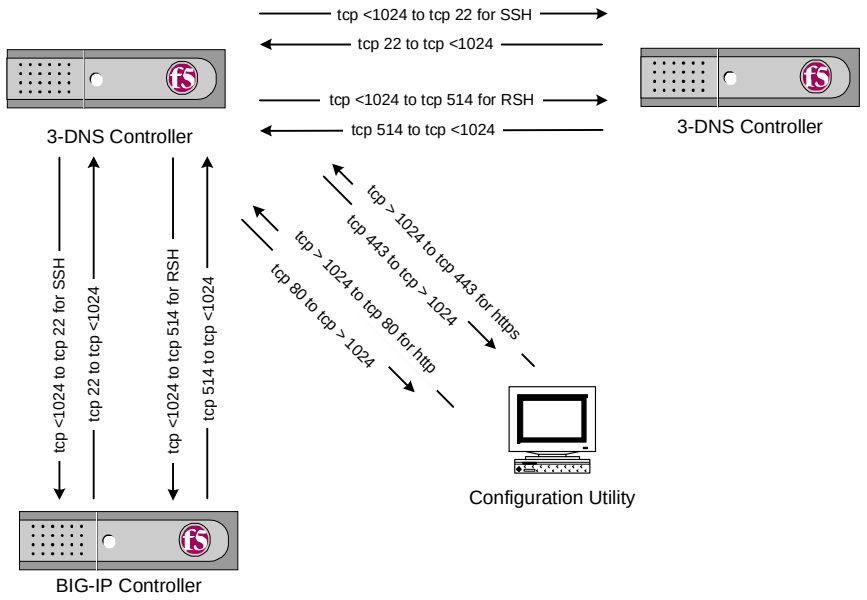


Figure 2.1 *Administrative communications*

Figure 2.2 shows the ports necessary for iQuery communication between 3-DNS Controllers and **big3d** agents that run on 3-DNS or BIG-IP Controllers.

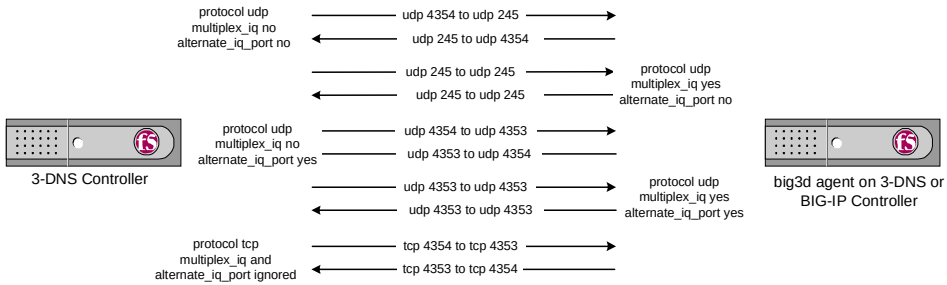


Figure 2.2 *iQuery communication*

Figure 2.3 shows the ports necessary for path probing between 3-DNS Controllers and LDNS servers.

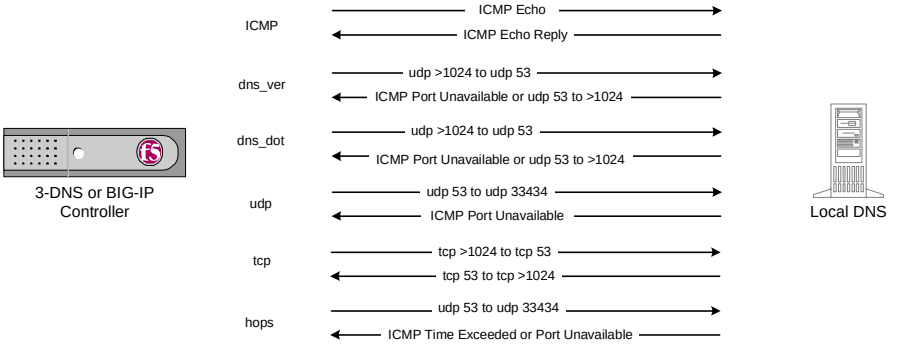


Figure 2.3 *Path probe communications*

Setting up communication between 3-DNS Controllers

The 3-DNS Controllers need to communicate with each other in order to synchronize configuration and performance data. If you use exclusively crypto 3-DNS Controllers (those which use **ssh** and **scp**), or exclusively non-crypto 3-DNS Controllers (those which do not use **ssh** and **scp**), 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**.

If you work in a mixed environment where some 3-DNS Controllers are crypto, and other 3-DNS Controllers are non-crypto, you need to enable the **rsh** and **rcp** tools on the crypto 3-DNS Controllers. Though the **rsh** and **rcp** tools come pre-installed on the crypto 3-DNS Controllers, you must explicitly enable these tools on the crypto 3-DNS Controllers. You can easily do this by running the **rsetup** script or the **config_rshd** script from the command line, or you can enable the tools when you run the First-Time Boot utility.

Table 2.2 shows the ports and protocols that 3-DNS Controllers use to communicate with each other.

From	To	Protocol	From Port	To Port	Purpose
Crypto 3-DNS Controller	Crypto 3-DNS Controller	tcp	<1023	22	SSH/SCP
Non-crypto 3-DNS Controller	Non-crypto 3-DNS Controller	tcp	>1024	514	RSH/RCP
Crypto 3-DNS Controller	Non-crypto 3-DNS Controller	tcp	>1024	514	RSH/RCP
Non-crypto 3-DNS Controller	Crypto 3-DNS Controller	tcp	>1024	514	RSH/RCP

Table 2.2 *Communications between 3-DNS Controllers*

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, 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 as described following.

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 by changing the **bigip.open_rsh_ports** system control variable to **0** in **/etc/rc.sysctl**.*

Table 2.3 shows the ports and protocols that 3-DNS Controllers use to communicate with BIG-IP Controllers. For more information about using **rsh** and **rcp** tools on the BIG-IP Controller, refer to the ***BIG-IP Controller Administrator Guide***.

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 2.3 *Communications between 3-DNS Controllers and BIG-IP Controllers*

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.

Table 2.4 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	4353 or 4354	4353	Alternate iQuery port for outbound traffic (open this port only when the **use_alternate_iq** global variable is set to **yes**)
big3d agent	3-DNS Controller	udp	245 or 4353	4354	Ephemeral port used for inbound iQuery traffic (when **multiplex_iq** is set to **no**)
big3d agent	3-DNS Controller	udp	245	245	Single port used for multiplexed inbound iQuery traffic (open this port only when the **multiplex_iq** global variable is set to **yes**)
big3d agent	3-DNS Controller	udp tcp	4353 4354 4353	4353 4353 4354	Single port used for multiplexed inbound iQuery traffic (open this port only when both the **use_alternate_iq** and the **multiplex_iq** global variables are set to **yes**)
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 2.4 *Communications between 3-DNS Controllers,* ***big3d*** *agents, and host servers*

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.

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 2.4, 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.

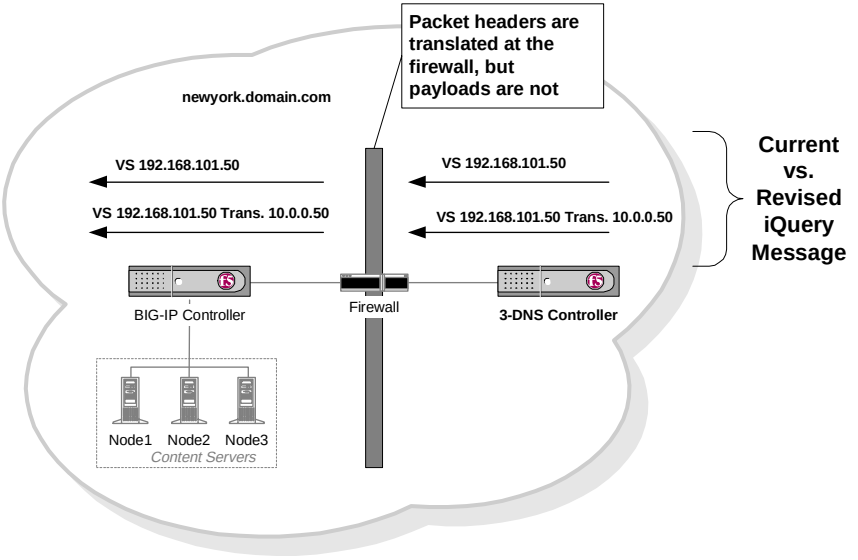


Figure 2.4 *Translating packet address through the firewall*

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 2.5 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 2.5 *Communications between 3-DNS Controllers and remote workstations*

Table 2.6 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 2.6 *DNS communications on the 3-DNS Controller*

Table 2.7 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	**big3d** agent	tcp	53	2000-12000	Probing using TCP (CISCO routers should "allow establish")
big3d agent	LDNS	udp	2000-12000	33434	UDP probing and traceroute
LDNS	**big3d** agent	icmp	N/A	N/A	Replies from ICMP, UDP pings, or traceroute probes
big3d agent	LDNS	dns_ver dns_dot	>1024	53	DNS version or dot queries
LDNS	**big3d** agent	dns_ver dns_dot	53	>1024	DNS version or dot response

Table 2.7 *Communication between **big3d** agents and LDNS servers*

The **big3d** agent can run on either a 3-DNS Controller or a BIG-IP Controller. 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**.

Planning issues for the load balancing configuration

The final phase of installing 3-DNS Controllers is setting up the load balancing configuration. Load balancing configurations are based on pools of virtual servers. When the 3-DNS Controller receives a connection request, it uses a load balancing mode to determine which virtual server in a given pool should receive the connection. The virtual servers in the pool can be the virtual servers managed by BIG-IP Controllers, or virtual servers managed by a generic host server, or they can be individual host servers themselves. Note that the 3-DNS Controller continuously verifies which virtual servers in the pool are currently available to accept load balanced connections.

Simple configurations typically use a single pool of virtual servers and a load balancing mode, such as Round Robin or Hops, that does not require significant additional configuration steps. More advanced load balancing configurations can use multiple pools, customizable load balancing modes, and other advanced traffic control features, such as topology-based access control and production rules. If you plan on implementing a more complex configuration, you may want to refer to Chapter 6 for additional details about advanced load balancing features.

Understanding the wide IP key

The ***wide IP key*** is the same address as the domain name. The wide IP key binds the information from DNS to the 3-DNS Controller, and indicates to DNS that the 3-DNS Controller (within the **named** process) should attempt to handle requests to this domain name. This allows the 3-DNS Controller to resolve the request by making a decision based upon its metric database and returning a better answer. Each wide IP definition must have its own unique address.

The wide IP key is sometimes referred to as the fallback address. When the preferred, alternate, and fallback load balancing modes (as specified in the wide IP definition) fail, the 3-DNS Controller instructs the DNS to issue its original answer. When this happens, the wide IP key is called the fallback address.

Choosing a load balancing mode

The 3-DNS Controller offers several different load balancing modes. Basic, static modes base load balancing on a pre-defined distribution pattern. More advanced, dynamic modes base load balancing on current network information such as the round trip time between a requesting client and a web server.

Basic load balancing

Basic load balancing modes distribute connections based on pre-defined distribution patterns, and do not take current server or network performance into account. The 3-DNS Controller supports the following basic load balancing modes:

- **Static Persist**
  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. 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**
  Round Robin mode distributes connections evenly across all virtual servers, passing each new connection to the next virtual server in line.
- **Ratio**
  Ratio mode distributes connections across virtual servers in proportion to a user-defined ratio. The distribution of replies is weighted Round Robin. For example, if one virtual server runs on a new, high-speed machine, and two other virtual servers run on older machines, you could set the ratio so that the high-speed virtual server receives twice as many connections as either of the two older virtual servers.
- **Global Availability**
  Global Availability mode distributes connections to a list of virtual servers, always sending a connection to the first available virtual server on the list.
- **Random**
  Random mode distributes connections in a random pattern.
- **Topology**
  Topology allows you to direct or restrict traffic flow by entering network information into the configuration file. This allows you to develop proximity-based mapping. For example, customers in a particular geographic region can be sent to virtual servers within that same region. The 3-DNS Controller determines the proximity of virtual servers by comparing the client's LDNS IP address to the IP address of the available virtual servers.

Advanced load balancing

Advanced load balancing bases connection distribution on current server and network performance information gathered by the **big3d** agent. The different dynamic load balancing modes incorporate different performance factors.

- **Quality of Service**
  Quality of Service (QOS) mode takes a variety of performance factors into account. You can configure the QOS mode to rate different performance factors higher or lower than others, or you

can configure the QOS mode to treat all factors as being equally important. The quality of the service equation calculates a performance score based on the following factors:

- Round trip time between the virtual server and the client LDNS
- Number of intermediate systems transitions between the virtual server and the client LDNS
- Number of packets currently being processed
- Percentage of packets completed
- Topological distribution
- Number of nodes currently **up** in the virtual server

❖ **Round Trip Times**
Round Trip Times mode sends each new connection to the virtual server that demonstrates the best round trip time between the virtual server and the client LDNS.

❖ **Hops**
Hops mode sends each new connection to the virtual server that has the fewest number of intermediate systems transitions between the virtual server and the client LDNS.

❖ **Packet Rate**
Packet Rate mode sends each new connection to the virtual server that has the least amount of network traffic.

❖ **Completion Rate**
Completion Rate mode sends each new connection to the virtual server that has the fewest number of dropped packets.

❖ **Least Connections**
Least Connections mode sends each new connection to the virtual server that currently hosts the fewest current connections. Note that you can use Least Connections mode only to load balance virtual servers managed by BIG-IP Controllers.

❖ **VS Capacity mode**
VS Capacity mode sends each new connection to the virtual server which has the most nodes **up**.

Ensuring availability for e-commerce, FTP, and other services that use multiple ports

Before the 3-DNS Controller selects a virtual server to receive a connection, it verifies that the virtual server is up and available. Certain types of network traffic, such as FTP traffic or e-commerce traffic, require that more than one port be available in order for the client's requests to be properly handled. For example, FTP servers use both ports 20 and 21, while e-commerce sites typically require that both ports 80 and 443 are available to handle HTTP and SSL traffic.

When you set up a load balancing configuration, you can define a list of ports that are verified on each virtual server before the virtual server is made available to receive load balanced connections.

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 advanced traffic control features

The 3-DNS Controller offers three advanced features that you can configure to further control the distribution and flow of network traffic.

- **Topology-based access control**
  With topology-based access control, you can restrict clients from connecting to virtual servers in specific data centers.

- **IP packet filtering**
  You can use IP packet filtering on the 3-DNS Controller to reject connection requests from certain source IP addresses.
- **Production rules**
  Use the production rules feature to change the load balancing configuration, as well as other system settings, based on dynamic factors such as current network performance, and time of day.

Configuring topology-based access control

Topology-based access control limits users to a subset of available servers based on their proximity to the servers. You can use topology-based access control to force European clients to connect only to servers also in Europe, rather than connecting to servers in South America.

The 3-DNS Controller uses topology-based access control in conjunction with load balancing modes. For example, even though topology-based access control may force a European client to connect to a server in Europe, the 3-DNS Controller still uses the load balancing modes specified for the preferred, alternate, and fallback methods to choose the best European server that should receive the connection.

For details about working with topology-based features and creating the necessary topology records, see Chapter 6, *Configuring Specialized Load Balancing*.

Setting up IP packet filtering

You can use the IP packet filtering feature to reject unwanted connection requests that may bog down your site and compromise performance. For example, if you detect an attack on your site, such as when your system is suddenly flooded with DNS requests from a single client, you can block that client's connection requests by defining an IP filter that rejects all packets containing that client's source IP address.

Note that the IP packet filtering supported on the 3-DNS Controller is based on BSD IP packet filtering. You can configure IP filters manually, but we recommend that you define IP filters in the Configuration utility, which greatly simplifies the process. To define a new IP filter in the Configuration utility, you specify one of three filter criteria:

- Source IP address only
- Destination IP address only
- Combination of source and destination IP address

Note that you can use IP address ranges when you define IP filters. You can find details about how to configure IP filters in Chapter 7, *Monitoring and Administration*.

Defining production rules

Production rules are a policy-based management feature that you can use to dynamically change the load balancing configuration and the system settings based on specific triggers, such as the time of day, or the current network traffic flow. You can set up standard production rules using the Configuration utility, or you can define custom production rules using the production rules scripting language.

Refer to Chapter 7, *Monitoring and Administration*, for information about setting up production rules.

Planning DNS zone file management

An important part of installing 3-DNS Controllers in your network is planning which servers should be master for a given DNS zone. When you initially set up a 3-DNS Controller in your network, you have two basic options for setting up DNS zone masters:

- You can use the 3-DNS Controller as the master DNS server for your domain.

- You can use an existing master DNS for your domain, and use the 3-DNS Controller only as the master DNS for your wide IP sub-domains.

The 3-DNS Controller must always be the DNS master for your wide IP sub-domains, regardless of which server is the master DNS in your network. We strongly recommend that you set up the 3-DNS Controller to run as a master DNS that manages your domain.

One major benefit of setting up the 3-DNS Controller to be the master DNS for your domain is that you can easily manage DNS zone files using NameSurfer, a browser-based, third-party application included on the 3-DNS Controller. You can also easily transfer your existing zone files to the 3-DNS Controller after the initial installation.

When you define wide IPs in the Configuration utility, the NameSurfer application automatically makes the appropriate additions to the zone files, and broadcasts the new zone files to the other DNS servers in your network. If you configure wide IPs manually, however, you need to make the corresponding zone file changes manually.

If you use the advanced synchronization features of the 3-DNS Controller, we strongly recommend that you configure each 3-DNS Controller to run as a master DNS. This effectively creates a group of peer/master DNS servers. This type of configuration offers the following advantages:

- You can change zone files on any one of the 3-DNS Controllers in the network and have those changes automatically broadcast to all of the other controllers in the network.
- Each 3-DNS Controller has the most up-to-date zone files, providing you one or more layers of redundancy.
- The NameSurfer application automatically controls the addition, configuration, and deletion of zone files.

Replacing your DNS servers with 3-DNS Controllers as master DNS servers for your domain

Figure 2.5 shows an implementation where both 3-DNS Controllers in the network act as master DNS servers for the domain, **domain.com**.

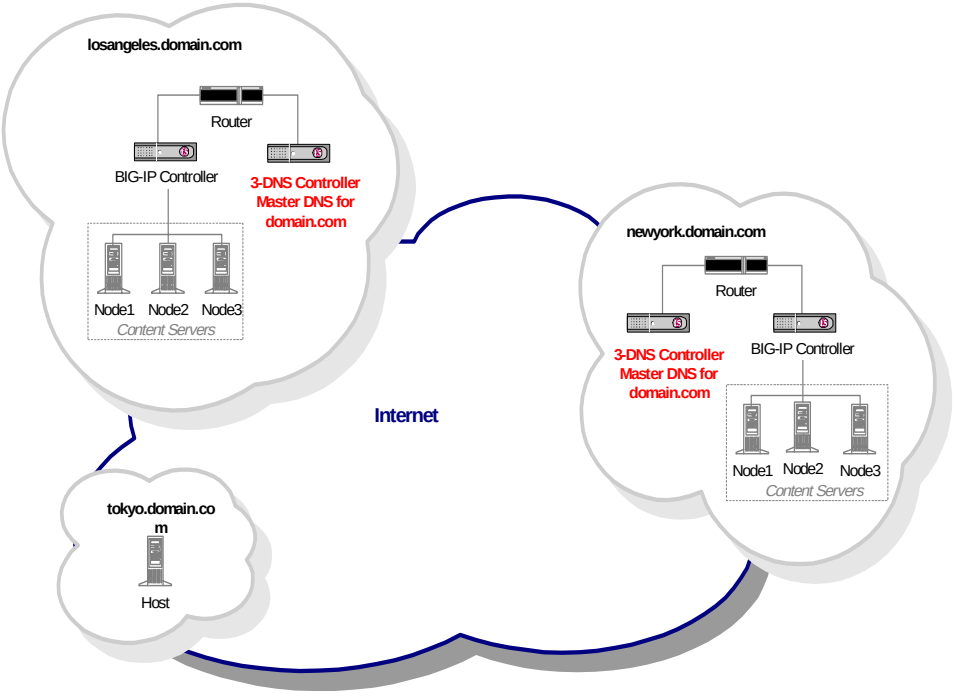


Figure 2.5 *Using 3-DNS Controllers as DNS masters for your domain*

Converting existing BIND files during an initial installation

After you initially install the 3-DNS Controller, you need to transfer existing BIND files, and then convert them to the NameSurfer format.

One option for converting your existing BIND files is to skip the NameSurfer configuration when you run the First-Time Boot utility. You transfer the zone files and **named.conf** file after the system has rebooted, and then run the **config_namesurfer** script that configures, converts, and starts the NameSurfer application. For more information, see *Appendix C, BIND 8 Information*.

The second option for importing your existing BIND files is to zone transfer from your current name server to NameSurfer. After configuring NameSurfer during the First-Time Boot utility and connecting to the Configuration utility, use the **Copy from other name server** option in the NameSurfer UI. For more information, refer to the NameSurfer documentation available from the splash screen in the Configuration utility.

Running 3-DNS Controllers as DNS masters for only wide IP sub-domains

At a minimum, all 3-DNS Controllers must be the DNS masters for the zones associated with wide IP definitions. When you set up a configuration where the 3-DNS Controllers are DNS masters for only those sub-domains, you need to make a few changes to the zone files on the master DNS for your domain after you configure the 3-DNS Controller.

Figure 2.6 shows an example where both 3-DNS Controllers are DNS masters for the wide IP sub-domains, and a generic name server in the Tokyo data center is the master DNS server for the domain, **domain.com**.

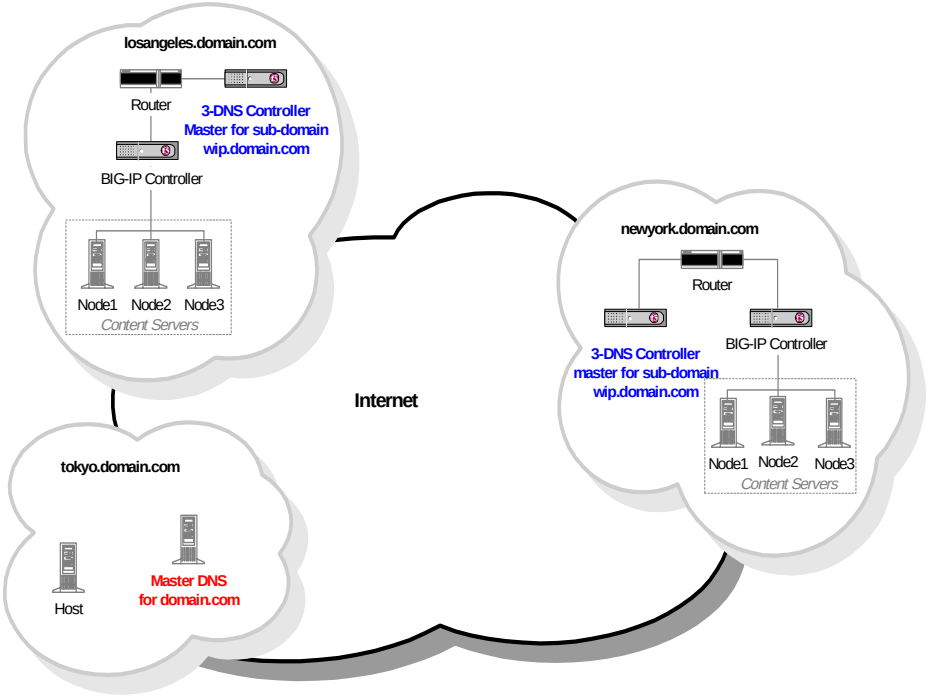


Figure 2.6 *The 3-DNS Controllers managing sub-domains*

Note that you should still set NameSurfer to be the master during the First-Time Boot utility for initial installations, or during the NameSurfer configuration script for upgrade installations. Remember that NameSurfer is the master for the zone files on the 3-DNS Controller, but in this configuration the zone files contain only those records associated with wide IPs configured on the 3-DNS Controller. When you configure wide IPs in the Configuration utility, the NameSurfer application automatically updates the corresponding sub-domain zones and broadcasts them to the other DNS servers in the network. For configurations where synchronization is enabled, changes to any NameSurfer files are automatically updated to the other 3-DNS Controllers.

F5 Networks
3·DNS®
Distributed Traffic Controller

3

Setting Up the Hardware

- Unpacking and installing the hardware
- Running the First-Time Boot utility
- Enabling remote login tools
- Preparing workstations for command line access

Unpacking and installing the hardware

There are two basic tasks you must complete to get the 3-DNS Controller installed and set up.

- Connect the peripheral hardware and connect the 3-DNS Controller to the network.
- Turn the system on and run the First-Time Boot utility. The First-Time Boot utility is a wizard that helps you configure basic system elements such as administrative passwords, IP addresses, and host names for both the root system and the 3-DNS web server. Once you complete the First-Time Boot utility, you can continue the configuration process either from a remote administrative workstation, or directly from the console.

Reviewing the hardware requirements

The 3-DNS Controller comes with the hardware you need for installation and maintenance. However, you must provide standard peripheral hardware, such as a keyboard or serial terminal.

Hardware provided with the 3-DNS Controller

When you unpack the 3-DNS Controller, make sure the following components are included:

- One power cable
- One PC/AT-to-PS/2 keyboard adapter
- Four rack-mounting screws
- Two keys for the front panel lock
- One extra fan filter
- One ***3-DNS Controller Administrator Guide***

If you purchase a hardware-based redundant system, you also receive one fail-over cable to connect the two controller units together (network-based redundant systems do not require a

fail-over cable). Additionally, if you purchase a 3-DNS Controller that supports encryption, you receive the ***F-Secure SSH Client*** manual, published by Data Fellows.

Peripheral hardware that you provide

For each 3-DNS Controller in the system, you need to provide the following peripheral hardware:

- Standard input/output hardware for direct administrative access to the 3-DNS Controller. Either of the following options is acceptable:
  - VGA monitor and PC/AT-compatible keyboard
  - Serial terminal and a null modem cable (See *To configure a serial terminal in addition to the console*, on page 3-10, for serial terminal configuration information.)
- Network hubs, switches, or concentrators to connect to the 3-DNS Controller network interfaces. The devices you select must be compatible with the network interface cards installed in the 3-DNS Controller. The devices can support 10/100 Ethernet, Gigabit Ethernet, or FDDI/CDDI (including multiple FDDI and full duplex).
  - For Ethernet, you need either a 10Mb/sec or 100 Mb/sec hub or switch
  - For FDDI/CDDI, a concentrator or a switch is optional

If you plan on doing remote administration from your own PC workstation, as most users do, we recommend that you have your workstation already in place. Keep in mind that the First-Time Boot utility prompts you to enter your workstation's IP address when you set up remote administrative access.

Familiarizing yourself with the 3-DNS Controller hardware

The 3-DNS Controller is offered in two hardware configurations: the 4U hardware configuration, and the 2U hardware configuration. Before you begin to install the 3-DNS Controller, you may want to

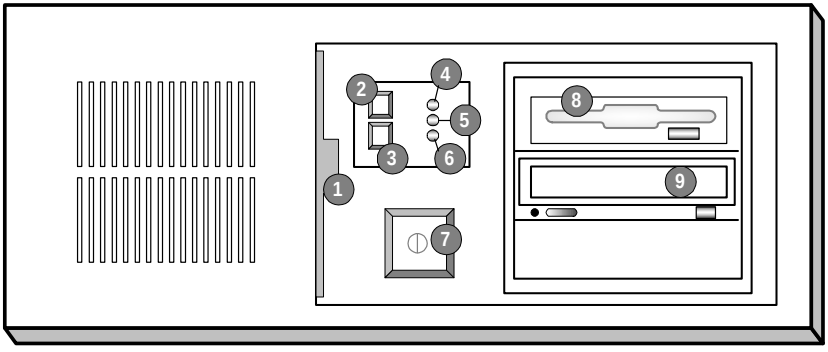
quickly review the following figures that illustrate the controls and ports on both the front and the back of a 4U 3-DNS Controller and a 2U 3-DNS Controller.

Using the 3-DNS Controller 4U hardware configuration

This section describes the front and back layout of a 4U 3-DNS Controller. If you have a special hardware configuration, such as those that include more than two interface cards, the ports on the back of your unit will differ slightly from those shown below.

◆ Note

The ports on the back of every 3-DNS Controller are individually labeled.

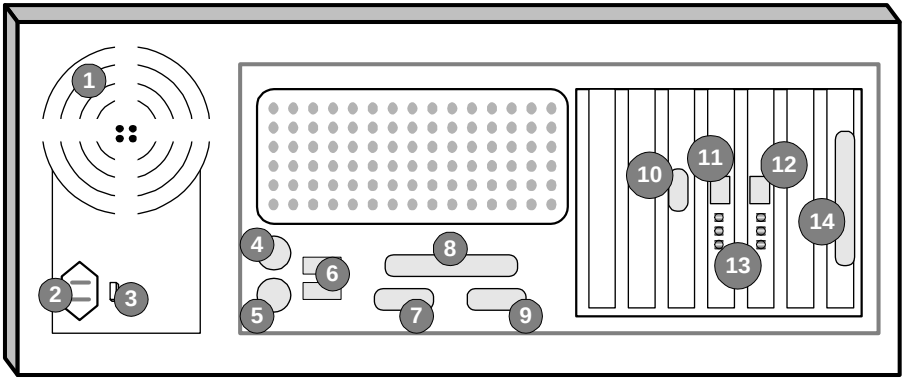


1. Fan filter	6. Power LED
2. Keyboard lock	7. On/off button
3. Reset button	8. 3.5 floppy disk drive
4. Keyboard lock LED	9. CD-ROM drive
5. Hard disk drive LED	

Figure 3.1 *Front view of a 4U 3-DNS Controller*

Figure 3.1 illustrates the front of a 4U 3-DNS Controller with the access panel open. On the front of the unit, you can turn the unit off and on, and you can reset the unit. You can also view the indicator lights for hard disk access and for the keyboard lock.

Figure 3.2 illustrates the back of a 4U 3-DNS Controller. Note that all ports are labeled, even those which are not intended to be used with the 3-DNS Controller. Ports marked with an asterisk (*) in the list following the figure are not used by the 3-DNS Controller, and you do not need to connect them to any peripheral hardware.



1. Fan	8. Printer port*
2. Power in	9. Fail-over port
3. Voltage selector	10. Video (VGA) port
4. Mouse port*	11. Internal interface (RJ-45)
5. Keyboard port	12. External interface (RJ-45)
6. Universal serial bus ports*	13. Interface indicator LEDs
7. Serial terminal port	14. Watchdog card*

Figure 3.2 *Rear view of a 4U 3-DNS Controller*

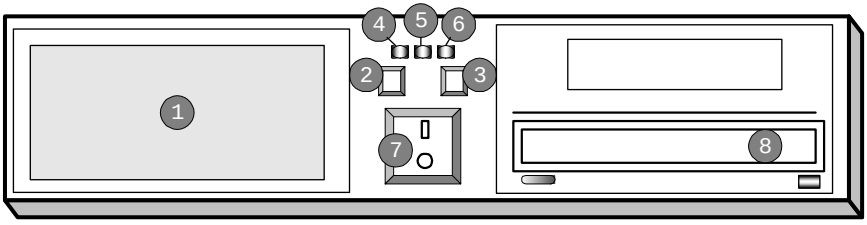
**Not to be connected to any peripheral hardware*

Using the 3-DNS Controller 2U hardware configuration

This section describes the front and back layout of a 2U 3-DNS Controller. If you have a special hardware configuration, such as those that include more than two interface cards, the ports on the back of your unit will differ slightly from those shown below.

◆ Note

The ports on the back of every 3-DNS Controller are individually labeled, so it should be clear what each port is, no matter which hardware configuration you have purchased.

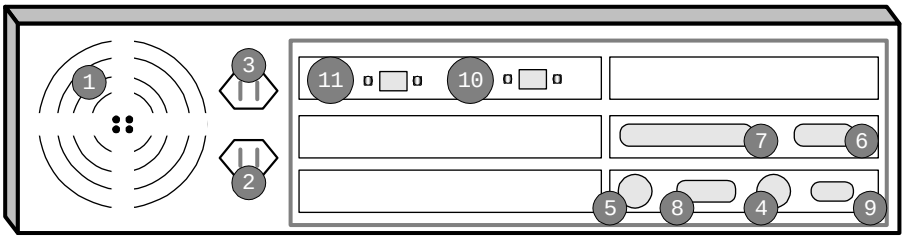


1. Fan filter	6. Power LED
2. Keyboard lock	7. On/off button
3. Reset button	8. CD-ROM drive
4. Keyboard lock LED	
5. Hard disk drive LED	

Figure 3.3 *Front view of a 2U 3-DNS Controller*

Figure 3.3 illustrates the front of a 2U 3-DNS Controller with the access panel open. On the front of the unit, you can turn the unit off and on, and you can reset the unit. You can also view the indicator lights for hard disk access and for the keyboard lock.

Figure 3.4 illustrates the back of a 2U 3-DNS Controller. Note that all ports are labeled, even those which are not intended to be used with the 3-DNS Controller. Ports marked with an asterisk (*) in the list following the figure are not used by the 3-DNS Controller, and you do not need to connect them to any peripheral hardware.



1. Fan
2. Power in
3. Power out
4. Mouse port*
5. Keyboard port
6. Fail-over port
7. Printer port*
8. Serial terminal port
9. Video (VGA) port
10. External interface (exp0)
11. Internal interface (exp1)

**Not to be connected to any peripheral hardware.*

Figure 3.4 *Back view of a 2U 3-DNS Controller*

Environmental requirements and usage guidelines

A 3-DNS Controller is an industrial network appliance, designed to be mounted in a standard 19-inch rack. To ensure safe installation and operation of the unit:

- Install the rack according to the manufacturer's instructions, and check the rack for stability before placing equipment in it.
- Build and position the rack so that once you install the 3-DNS Controller, the power supply and the vents on both the front and back of the unit remain unobstructed. The 3-DNS Controller must have adequate ventilation around the unit at all times.
- Do not allow the air temperature in the room to exceed 40° C.

- Do not plug the unit into a branch circuit shared by more electronic equipment than the circuit is designed to manage safely at one time.
- Verify that the voltage selector is set appropriately before connecting the power cable to the unit.

The unit must be connected to Earth ground, and it should have a reliable ground path maintained at all times.

The 3-DNS Controller contains a lithium battery. There is danger of an explosion if you replace the lithium battery incorrectly. We recommend that you replace the battery only with the same type of battery originally installed in the unit, or with an equivalent type recommended by the battery manufacturer. Be sure to discard all used batteries according to the manufacturer's instructions.

This equipment is not intended for operator serviceability. To prevent injury and to preserve the manufacturer's warranty, allow only qualified service personnel to service the equipment.

Guidelines for DC powered equipment

A DC powered installation must meet the following requirements:

- Install the unit using a 20 Amp external branch circuit protection device.

- For permanently connected equipment, incorporate a readily-accessible disconnect in the fixed wiring.
- Use only copper conductors.

Install DC powered equipment only in restricted access areas, such as dedicated equipment rooms, equipment closets, or similar locations.

Installing and connecting the hardware

There are six basic steps to installing the hardware. You simply need to install the controller in the rack, connect the peripheral hardware and the external and internal interfaces, and then connect the fail-over and power cables. If you have a unit with three or more network interface cards (NICs), be sure to review step 3.

WARNING

Do not turn on a 3-DNS Controller until all peripheral hardware is connected to the unit.

To install the hardware

1. Mount the 3-DNS Controller on the rack and secure it using the four rack-mounting screws that are provided.
2. Connect the hardware that you have chosen to use for input/output:
   - If you are using a VGA monitor and keyboard, connect the monitor connector cable to the video port (number 10 in the 4U figure, or number 9 in the 2U figure) and connect the keyboard connector cable to the keyboard port (number 5 in the 4U or 2U figure). Note that a PC/AT-to-PS/2 keyboard adapter is included with each 3-DNS Controller (see the component list on page 3-1).

- Optionally, if you are using a serial terminal as the console, connect the serial cable to the serial terminal port (number 7 in the 4U figure, or number 8 in the 2U figure). You should not connect a keyboard to the 3-DNS Controller. If there is no keyboard connected to the 3-DNS Controller when it is started or rebooted, the 3-DNS Controller defaults to using the serial port as the console.

3. Connect the external interface (number 11 in the 4U figure, or number 10 in the 2U figure) to the network from which the 3-DNS Controller receives connection requests.

   - If you have purchased a unit with three or more network interface cards (NICs), be sure to note or write down how you connect the cables to the internal and external interfaces. When you run the First-Time Boot utility, it automatically detects the number of interfaces that are installed and prompts you to configure more external interfaces, if you want. It is important to select the correct external interface based on the way you have connected the cables to the back of the unit.

4. Connect the internal interface (number 11 in the 4U or 2U figure) to the network that houses the array of servers, routers, or firewalls that the 3-DNS Controller load balances.

5. If you have a hardware-based redundant system, connect the fail-over cable to the serial terminal port on each unit (number 7 in the 4U figure, or number 8 in the 2U figure).

6. Connect the power cable to the 3-DNS Controller (number 2 in the 4U or 2U figure), and then connect it to the power source.

WARNING

Before connecting the power cable to a power supply, customers outside the US should make sure that the voltage selector is set appropriately. This check is necessary only if the controller has an external voltage selector,

To configure a serial terminal in addition to the console

To configure a serial terminal, in addition to the standard console, for the 3-DNS Controller, you need to complete the following configuration steps. Note that if you are using a serial vt100 connection, you must edit both the **/etc/ttys** and **bash_profile** files on the 3-DNS Controller.

◆ Note

Before you configure the serial terminal, you must disconnect the keyboard from the 3-DNS Controller. When there is no keyboard connected to the 3-DNS Controller, the 3-DNS Controller defaults to using the serial port for the console.

You must attach a serial device to the serial port before the 3-DNS Controller is booted in order for the controller to use the serial port as the console.

1. Configure the serial terminal settings as follows:

   -9600 baud

   -8 bits

   -1 stop bit

   -No parity

2. Open the **/etc/ttys** file and find the line that reads **tty00 off**. Modify it as shown here:

   ```
   # PC COM ports (tty00 is DOS COM1) tty00
   "/usr/libexec/getty default" vt100 in secure
   tty01 off
   ```

3. Save the **/etc/ttys** file and close it.

4. Reboot the BIG/ip Controller.

Running the First-Time Boot utility

After you have finished connecting the 3-DNS Controller and peripheral hardware to the network, you then run the First-Time Boot utility. The First-Time Boot utility is a wizard that walks you through a brief series of required configuration tasks, such as defining a root password and configuring IP addresses for the network interfaces. Once you complete the First-Time Boot utility, you can connect to the 3-DNS Controller from a remote workstation and begin configuring your load balancing set up.

The First-Time Boot utility is organized into three phases: configure, confirm, and commit. You first configure all of the required information. Next, you have the opportunity to correct, if necessary, and confirm each individual setting that you have configured. Finally, your confirmed settings are committed and saved to the system.

Each phase walks you through a series of screens, presenting the information in the following order:

- Root password
- Host name
- Default route (typically a router's IP address)
- Time zone
- NTP clocks
- Interface settings for the network interface(s)
- Configuration for 3-DNS redundant systems (fail-over IP address)
- IP address for remote administration
- Settings for the 3-DNS web server
- Defining the basic BIG-IP and sync group configuration
- Settings for the NameSurfer application
- Allowance of technical support access

The screens you see are tailored to your specific hardware and software configuration. For example, if you have a stand-alone system, the First-Time Boot utility skips the redundant system screens.

Gathering the information

Before you run the First-Time Boot utility on a 3-DNS Controller, you should have the following information ready to enter:

- Passwords for the root system, for the 3-DNS web server, and for technical support access (optional)
- Host names for the root system and the 3-DNS web server
- A default route (typically a router's IP address)
- Settings for the network interfaces, including IP addresses, media type, and custom netmask and broadcast addresses
- Configuration information for redundant systems, including an IP alias for the shared address, and the IP addresses of the individual controllers
- The IP address or IP address range for remote administrative connections
- The IP addresses of the other 3-DNS Controllers and BIG-IP Controllers running in the network

An important note about configuring international 3-DNS Controllers

When you run the First-Time Boot utility on a non-crypto 3-DNS Controller, certain screens are different from those shown when you run the First-Time Boot utility on a crypto 3-DNS Controller.

- On crypto 3-DNS Controllers, the First-Time Boot utility prompts you to configure an administrative IP address from which the 3-DNS Controller accepts **ssh** connections.
- On non-crypto 3-DNS Controllers, the First-Time Boot utility prompts you to configure an administrative IP address from which the 3-DNS Controller accepts **rsh** connections.

The 3-DNS Controller stores the administrative IP address for **rsh** and **rcp** connections in the **/etc/hosts.allow** file. Note that storing the administrative IP address in the **/etc/hosts.allow** file may differ slightly from other common **rsh** configurations where it is often stored in the **/etc/hosts.equiv** file.

Starting the First-Time Boot utility

The following steps get you started with the First-Time Boot utility. As you work through the First-Time Boot utility, refer to the following sections that provide you with important information regarding each screen that the First-Time Boot utility presents.

Run the First-Time Boot utility directly on the console, using the VGA monitor and keyboard.

1. Turn on the power switch.

   Once you turn on the power switch (located on the front of the 3-DNS Controller as shown in Figure 3.1, number 7), the License Agreement screen appears.

2. Read the License Agreement as you page down to the end of the License Agreement screen, and press **Enter**.

3. If you accept the agreement, select **Yes, I Agree To This License,** and press **Enter**.
   The Welcome screen appears.

4. Continue to press any key until you come to the New Root Password screen.

5. Each screen in the First-Time Boot utility provides instructions on how to proceed. For additional information on how to fill in the remaining screens in the First-Time Boot utility, read the following sections.

Defining a root password

A root password allows you administrative access to the 3-DNS Controller. The password must contain a minimum of 6 characters, and a maximum of 128 characters. Passwords are case-sensitive,

and we recommend that your password contain a combination of uppercase and lowercase characters, as well as punctuation characters. Once you enter a password, the First-Time Boot utility prompts you to confirm your root password by typing it again. If the two passwords match, your password is immediately saved. If the two passwords do not match, you receive an error message asking you to re-enter your password.

WARNING

The root password is the only setting that is saved immediately; the other settings are confirmed and committed at the end of the First-Time Boot utility process. You can change the root password after the First-Time Boot utility completes and you reboot the 3-DNS Controller (see To change the root user password for command line access, on page 7-11 for details). You can change other system settings when the First-Time Boot utility prompts you to confirm your configuration settings.

Defining a host name

The host name identifies the 3-DNS Controller. Host names must start with a letter or number and contain at least two characters. They may contain numbers, letters, dash symbols (**-**), and periods (**.**). There are no additional restrictions on host names, other than those imposed by your own network requirements.

Configuring a default route

If a 3-DNS Controller does not have a predefined static route for network traffic, the unit automatically sends traffic to the IP address that you define as the default route. Typically, a default route is set to a router's IP address.

Configuring a time zone

Configuring a time zone ensures that the clock for the 3-DNS Controller is set correctly, and that dates and times recorded in log files correspond to the time zone of the system administrator. Scroll through the time zone host to find the zone closest to your location. Note that one option may appear with multiple names.

Configuring NTP clocks

You can synchronize your time to a public time server by using Network Time Protocol (NTP). NTP is built on top of TCP/IP and assures accurate local timekeeping with reference to clocks located on the Internet. This protocol is capable of synchronizing distributed clocks within milliseconds over long periods of time. If you choose to do this, make sure UDP port 123 is open in both directions when 3-DNS is behind a firewall.

Configuring the interfaces

If you have a redundant system, on the Configure 3-DNS Interfaces screen, select **Yes, it is a redundant 3-DNS System**. You must configure the primary Ethernet interface, but you configure the secondary Ethernet interface only if you want to have two independent network access paths to the 3-DNS Controller. The utility prompts you for each interface, and asks you to provide the IP address, netmask, broadcast address, and the interface media type.

If you have a redundant system, you are also prompted to provide the IP address that serves as an IP alias for both 3-DNS Controllers. The IP alias is shared between the units, and is used only by the currently active machine. Each individual controller uses unique IP addresses on its network interface card(s). The First-Time Boot utility guides you through configuring the interfaces, based on your hardware configuration.

- **Stand-alone controllers**
  On stand-alone controllers, you enter IP addresses in the following order: primary Ethernet interface IP address, secondary Ethernet interface IP address.
- **Redundant systems**
  On redundant systems, you enter IP addresses in the following order: primary Ethernet interface IP address, primary shared alias, secondary Ethernet interface IP address, secondary shared alias.

Configuring the primary and secondary Ethernet interfaces

The Select Interface screen shows a list of the installed interfaces. Select the Ethernet interface you want to configure, and press **Enter** (the primary Ethernet interface is typically named **exp0**). The utility prompts you for the following information, in many cases offering you a default:

- Interface IP address
- Netmask
- Broadcast address
- Primary shared IP alias (redundant systems only)
- Primary shared alias netmask (redundant systems only)
- Primary shared alias broadcast address (redundant systems only)
- Interface media type
- Peer IP address (redundant systems only)

Note

The IP address of the primary Ethernet interface is not the IP address associated with your domain(s). The IP addresses of the domains themselves are specified by the wide IP definitions.

WARNING

The First-Time Boot utility lists only the network interface cards that it detects during boot up. If the utility lists only one interface card, the network adapter may have come loose during shipping.

Check the LED indicators on the network adapters to ensure that they have properly detected the 3-DNS Controller media that should be installed.

Once you select the interface, you need to enter the following information:

- **IP address**
- **Netmask**
  Note that the 3-DNS Controller uses a default netmask appropriate to the subnetwork indicated by the IP address. The default netmask is shown in brackets at the prompt. To accept the default, press **Return**.
- **Broadcast address**
  The default broadcast address is a combination of the IP address and the netmask. The default broadcast address is shown in brackets at the prompt. To accept the default, press **Return**.
- **Primary shared IP alias (redundant systems only)**
- **Peer IP address (redundant systems only)**
  The peer IP address is the IP address of the other controller that runs in the redundant system. The 3-DNS Controller uses the specified peer IP address to communicate with the second controller.
- **Media type for primary Ethernet interface**
  The media type options depend on the network interface card included in your hardware configuration. The 3-DNS platform supports the following types:
  - Auto
  - 10baseT
  - 10baseT,FDX
  - 100baseTX
  - 100baseTX,FDX
  - FDDI
  - Gigabit Ethernet

Configuring remote administration

When you configure remote administration, the screens that you see vary, depending on whether you have a US 3-DNS Controller, or an international 3-DNS Controller.

- On a US 3-DNS Controller, the first screen you see is the Configure SSH screen, which prompts you to type an address for SSH command line access.
- On international 3-DNS Controllers that do not have SSH, the First-Time Boot utility displays the Configure rsh screen instead.

The First-Time Boot utility prompts you to enter a single IP address or a range of IP addresses, from which the 3-DNS Controller can accept administrative connections (either remote shell connections, or connections to the 3-DNS web server). To specify a range of IP addresses, you can use the asterisk (*****) as a wildcard character in the IP addresses.

The following example allows remote administration from all hosts on the **192.168.2.0** network:

```
192.168.2.*
```

◆ Tip

For redundant systems, you must configure command line access. If you do not configure command line access, the two controllers in the system cannot communicate with each other, and they cannot properly initiate a fail-over.

Configuring settings for the 3-DNS web server

The 3-DNS web server requires that you define a domain name for the server on the primary Ethernet interface. If you are using the secondary Ethernet interface, you must define a domain name on that interface as well. The 3-DNS web server configuration requires that you define a user ID and password. On US products, the configuration also generates certificates for authentication.

The First-Time Boot utility guides you through a series of screens to set up web server access:

- The first screen prompts you to enter a fully-qualified domain name for each network interface that you configured earlier in the utility.
- The certification screen prompts you to enter the company name and location information used for the authentication certificate (US 3-DNS Controllers only).
- The next web server screen prompts you for a user name and a password. The password does not show on screen as you type it. The utility prompts you to enter the password again for confirmation purposes.
- The final screen prompts you to specify whether you want to allow F5 technical support to have access to the web server.

Note that if you ever change the IP addresses or host names on the 3-DNS Controller interfaces, you need to reconfigure the 3-DNS web server to reflect your new settings. You can run the re-configuration utility from the command line using the following command:

```
config_httpd
```

If you wish to create a new password for the 3-DNS web server, delete the **/var/f5/httpd/basicauth/users** file before running the **config-httpd** utility. If this file is missing from the configuration, the utility prompts you for both user ID and password information.

You can also add users to the existing password file, change a password for an existing user, or recreate the password file, without actually going through the 3-DNS web server configuration process. For more information, see *To add a new user ID using the Configuration utility*, on page 7-11.

WARNING

*If you have modified the 3-DNS web server configuration outside of the configuration utility, be aware that some changes may be lost when you run the **config-httpd** utility. This utility overwrites the **httpd.conf** file, and several other files, but it does warn you before doing so.*

Identifying additional controllers in the network

In the next series of screens, you identify other 3-DNS Controllers and BIG-IP Controllers running on the network that the current 3-DNS Controller needs to communicate with. For each additional 3-DNS Controller or BIG-IP Controller, you must enter the IP address, the root user ID, whether you access the controller using SSH or **rsh**, and you must select which sync group the controller belongs to. A ***sync group*** is a group of 3-DNS Controllers that share configuration settings and path statistics.

Note that if you are defining a redundant system (either 3-DNS Controllers or BIG-IP Controllers), you need to enter the IP address of the controller, as well as the shared IP alias for each interface on the controller. You also need to choose whether the current 3-DNS Controller will be a principal or a receiver (note that each sync group can have only one principal, but can have an unlimited number of receivers).

You can view the list of currently-defined controllers at any time, and you can use the **List is Complete** option to exit these screens and move on to the next configuration task.

Configuring the NameSurfer application for zone file management

In the final series of screens, you choose whether to have NameSurfer handle DNS zone file management on the current 3-DNS Controller. We strongly recommend that you configure NameSurfer to handle zone file management by selecting NameSurfer to be the master on the 3-DNS Controller. If you select NameSurfer as the master, NameSurfer converts the master DNS zone files on the controller and handles all changes and updates to these files. (You can access the NameSurfer application

directly from the Configuration utility). For details about converting existing BIND files to NameSurfer, see *To transfer and convert existing BIND files*, on page C-22.

WARNING

If you do not set NameSurfer to be the master for your wide IP zones, you cannot use the Configuration utility. Instead, you must manually configure all 3-DNS Controller settings.

Confirming your configuration settings

By this point, you should already have entered all the configuration information, and now you confirm each setting. Each confirmation screen displays a setting and prompts you to accept or edit it. If you choose to edit the setting, the utility displays the original configuration screen in which you defined the setting the first time. When you finish editing the item, you return directly to the Confirmation screen for that item, and continue the confirmation process. Note that once you accept a setting in the Confirmation screen, you do not have another opportunity to review it.

You confirm or edit the settings in the same order that you configured them:

- Confirm Host name
- Confirm Default route
- Confirm time zone
- Confirm all interface settings, external and internal
- Confirm administrative IP address
- Confirm web server options

Once you have confirmed the last setting, the First-Time Boot utility moves directly into the commit phase, where you are not able to make any changes.

Committing your configuration settings to the system

Once you confirm all of the configuration settings, the configuration utility saves the configuration settings. During this commit process, the First-Time Boot utility creates the following files and tables:

- An administrative IP access file
  This file stores the IP address, or IP address range, from which the 3-DNS Controller accepts administrative connections.
- An **/etc/wideip.conf** file
- An **/etc/netstart** file
- An **/etc/ethers** file
- A **/var/f5/httpd/conf/httpd.conf** file
- An **/etc/sshd_config** file

If you want to change any information in these files at a later time, you can edit the files directly, change the information in the web-based Configuration utility, or change certain settings using command line utilities. If necessary, you can also re-run the First-Time Boot utility.

Enabling remote login tools

If you are setting up a crypto 3-DNS Controller that needs to communicate with international 3-DNS Controllers, you must enable the **rsh** and **rcp** tools on the crypto 3-DNS Controller. These are the standard communication and copying tools that international 3-DNS Controller and BIG-IP Controllers use.

To enable the remote login tools on a US 3-DNS Controller

Run the **rsetup** script from the command line. The **rsetup** script performs several essential steps to enable access for **rsh** and **rcp**, and we strongly recommend that you use the script rather than doing this manually.

Preparing workstations for command line access

The type of system you have determines the options you have for remote command line administration:

- crypto 3-DNS Controllers support secure shell command line access using the F-Secure SSH client.
- non-crypto 3-DNS Controllers support command line access using a standard **rsh** shell.

If you are working with a crypto 3-DNS Controller, we recommend that you install the F-Secure SSH client on your workstation. The 3-DNS Controller includes a version of the F-Secure SSH client for each of the following platforms: Windows, UNIX, and Macintosh. You can download the F-Secure client using your web browser, or by using an FTP server on the administrative workstation.

The F-Secure license agreement allows you to download two copies of the F-Secure SSH client. If you require additional licenses, contact Data Fellows. For information about contacting Data Fellows, as well as information about working with the SSH client, refer to the F-Secure manual included with your 3-DNS Controller.

◆ Note

You can also use the F-Secure SSH suite to transfer files to and from the 3-DNS Controller, and for remote backups. An F-Secure SSH client is pre-installed on the 3-DNS Controller to assist with file transfer activities. Please refer to the F-Secure User's Manual for more information.

Downloading the F-Secure SSH client from the 3-DNS web server

The F-Secure SSH client is available in the Downloads section of the 3-DNS web server. For US products, you connect to the 3-DNS web server via SSL on port 443 (use **https://** rather than **http://** in

the URL). Once you connect to the 3-DNS web server, click the **Downloads** link. From the Downloads page, you can select the SSH Client.

Downloading the F-Secure SSH client using FTP

The 3-DNS Controller has an FTP client installed, which allows you to transfer the F-Secure SSH Client using FTP. (Note that your destination workstation must also have an FTP server installed.) After you transfer the installation file, you simply decompress the file and run the F-Secure installation program.

You initiate the transfer from the 3-DNS Controller itself using the monitor and keyboard or the serial terminal attached directly to the 3-DNS Controller.

To transfer the SSH client using FTP

1. Locate the appropriate SSH client for the operating system that runs on the administrative workstation:

   a) Navigate to the **/usr/contrib/fsecure** directory where the F-secure SSH clients are stored.

   b) List the directory, noting the file name that corresponds to the operating system of your administration workstation.

2. Start FTP:
   `ftp`

3. Open a connection to the remote workstation using the following command, where **IP address** is the IP address of the remote workstation itself:

   `open <IP address>`

   Once you connect to the administrative workstation, the FTP server on the administrative workstation prompts you for a password.

4. Type the appropriate user name and password to complete the connection.

5. Switch to passive FTP mode:

   `passive`

6. Switch the transfer mode to binary:

   `bin`

7. Navigate to the directory on the administrative workstation where you want to install the F-Secure SSH client.

8. Start the transfer process using the following command, where **filename** is the name of the F-Secure file that is specific to the operating system running on the administrative workstation:

   `put <filename>`

9. Once the transfer is done, type the following command:

   `quit`

Setting up the F-Secure SSH client on a Windows 95 or Windows NT workstation

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

To unzip and install the SSH client

1. Log on to the Windows workstation.

2. Navigate to the directory where you transferred the F-Secure installation file, and run PKZip or WinZip to extract the files.

3. The set of files extracted includes a Setup executable. Run the Setup executable and install the client.

4. Start the F-Secure SSH client.

5. In the SSH Client window, from the File menu choose **Connect**.
   The Connect Using Password Authentication window opens.

6. Click **Properties**.
7. In the Options dialog box, check **Compression** and **Forward X11**, and set the Cipher option to **Blowfish**. Click **OK** to return to the Connect Using Password Authentication window.
8. In the Connect Using Password Authentication window, type the following items:

   a) 3-DNS Controller IP address or host name

   b) The root user name

   c) The root password
9. Press Return to log on to the 3-DNS Controller.

Setting up the F-Secure SSH client on a UNIX workstation

The F-Secure installation file for UNIX platforms is compressed in TAR/Gzip format.

To untar and install the SSH client

1. Log on to the workstation and navigate to the directory where you transferred the F-Secure SSH client tar file.
2. Untar the file and follow the instructions in the **install** file to build the F-Secure SSH client for your workstation.
3. Start the SSH client.
4. Open a connection to the 3-DNS Controller:

   ```
   ssh -l root [3-DNS IP address]
   ```
5. Type the root password.

F5 Networks
3·DNS®
Distributed Traffic Controller

4

Defining the Network Setup

- Setting up a basic configuration
- Setting up a data center
- Setting up servers
- Setting up sync groups
- Configuring global variables
- Configuring IP filters
- Configuring Sendmail

Setting up a basic configuration

The second phase of installing 3-DNS Controllers is to define the network setup. Each 3-DNS Controller in the network setup must have information regarding which data center stores specific servers, and with which other 3-DNS Controllers it can share configuration and load balancing information. A basic network setup includes data centers, servers, wide IPs, and one sync group.

You can configure global variables that apply to all servers and wide IPs in your network. However, the default values of the global variables work well for most situations, so configuring global variables is optional. You can find more information about global variables in *Configuring global variables*, on page 4-33.

The following sections describe the various elements of a basic network:

- **Data centers**
  Data centers are the top level of your network setup. We recommend that you configure one data center for each physical location in your global network. The data center element of your configuration defines the servers (3-DNS Controllers, BIG-IP Controllers, and hosts) that reside at that location.

  A data center can contain any type of server. For example, in Figure 4.1, the Tokyo data center contains a 3-DNS Controller and a host, while the New York and Los Angeles data centers contain 3-DNS Controllers and BIG-IP Controllers.

  To configure data centers, see *Setting up a data center*, on page 4-2.

- **Servers**
  The servers that you define in the network setup include 3-DNS Controllers, BIG-IP Controllers, and host machines. You define the 3-DNS Controllers that manage the BIG-IP Controllers and hosts, and you also define the virtual servers that are managed by the BIG-IP Controllers and hosts. Virtual servers are the ultimate destination for connection requests.

  To configure servers, see *Setting up servers*, on page 4-5.

- **Sync groups**
  Sync groups contain only 3-DNS Controllers. When setting up a sync group, you define which 3-DNS Controllers have the same configuration. In most cases, you should define all 3-DNS Controllers as part of the same sync group.

  To configure sync groups, see *Setting up sync groups*, on page 4-30.

- **Wide IPs**
  After you define virtual servers for your BIG-IP Controllers and hosts, you need to specify how connections are distributed among the virtual servers by defining wide IPs. A wide IP maps a domain name to a pool of virtual servers, and it specifies the load balancing modes that the 3-DNS Controller uses to choose a virtual server from the pool.

  When an LDNS requests a connection to a specific domain name, the wide IP definition specifies which virtual servers are eligible to answer the request, and which load balancing modes to use in choosing a virtual server to resolve the request.

  To configure wide IPs, see *Adding a wide IP*, on page 5-12,

Setting up a data center

The first step in configuring your 3-DNS Controller network is to create data centers. A data center defines the group of 3-DNS Controllers, BIG-IP Controllers, and hosts that reside in a single physical location.

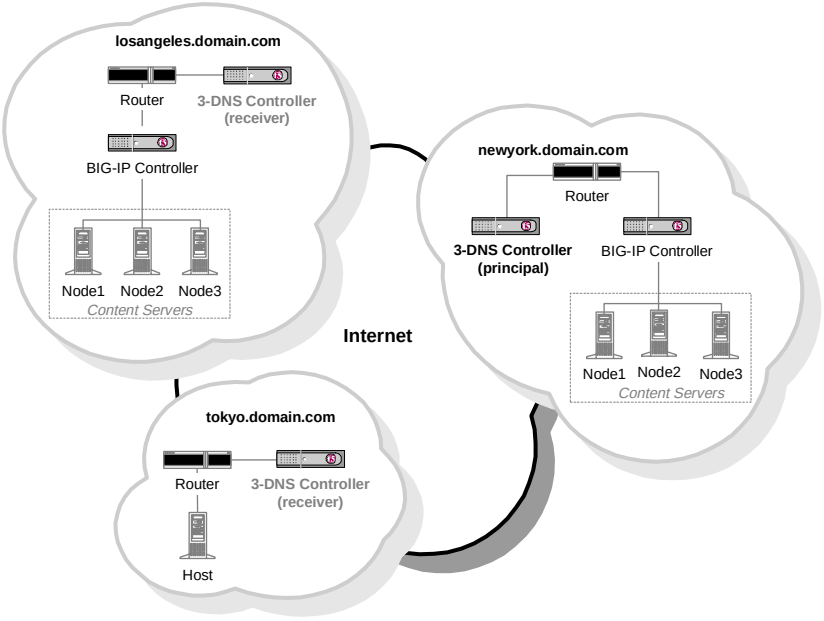


Figure 4.1 *Example network setup*

The advantage of grouping all machines from a single location into one data center is to allow path information collected by one machine to be shared with all other machines in the data center. For example, when a host machine belongs to a data center, the host can take advantage of the information collected by the **big3d** agent, which runs only on 3-DNS Controllers and BIG-IP Controllers. Without the information that the **big3d** agent collects, virtual servers owned by host machines would not be able to use advanced load balancing modes.

To configure a data center using the Configuration utility

1. In the navigation pane, click **Data Centers**.
2. On the toolbar, click **Add Data Center**.
   The Add New Data Center screen opens.

3. Add the new data center settings. For help on defining data centers, click **Help** on the toolbar.
   The data center is added to your configuration.
4. Repeat this process for each data center in your network.

When you add servers to the network setup, you assign the servers to the appropriate data centers.

To configure a data center manually

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. Select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
   An environment variable determines whether this command starts vi or pico.
3. Locate or add the **datacenter** statement.

   The **datacenter** statement should be the second statement in the file, after the **globals** statement and before **server** statements.
4. In the first line of the **datacenter** statement, type a name for the data center and enclose the name in quotation marks, as shown in Figure 4.2.
5. Type the server type and IP address for each server that is part of the specified data center.

   Figure 4.2 shows the correct syntax for the **datacenter** statement.

```
datacenter {
  name <"data center name">
  [ location <"location info"> ]
  [ contact <"contact info"> ]
  [ 3dns <3-DNS IP address> ]
  [ bigip <BIG-IP IP address> ]
  [ host <host IP address> ]
}
```

Figure 4.2 *Syntax for the datacenter statement*

Repeat the above procedure until you have added a separate **datacenter** statement for each data center on your network.

Figure 4.3 shows a sample **datacenter** statement.

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

Figure 4.3 *Sample data center definition*

Setting up servers

There are three types of servers: 3-DNS Controllers, BIG-IP Controllers, and other hosts. At the minimum, your network includes one 3-DNS Controller, and at least one server (BIG-IP Controller or host) that it manages.

This section describes how to set up each 3-DNS Controller, BIG-IP Controller, and host machine that make up your network. The setup procedures here assume that the BIG-IP Controllers and hosts are up and running, and that they already have virtual servers defined. Note that 3-DNS Controllers do not manage virtual servers.

Defining 3-DNS Controller servers

The purpose of defining a 3-DNS Controller server is to establish in which data center the 3-DNS Controller resides and, if necessary, to change **big3d** agent settings. In setting up a 3-DNS Controller server, you also make that 3-DNS Controller available so you can add it to a sync group.

To define a 3-DNS Controller server using the Configuration utility

1. In the navigation pane, click **3-DNS Controllers**.
2. On the toolbar, click **Add 3-DNS Controller**.
   The Add New 3-DNS Controller screen opens.
3. Add the new 3-DNS Controller settings. For help on defining 3-DNS Controllers, click **Help** on the toolbar.

The 3-DNS Controller is added to your configuration. Repeat this procedure for each 3-DNS Controller you need to add.

To define a 3-DNS Controller server manually

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Use the syntax shown in Figure 4.4 to define a 3-DNS Controller.

   All **server** statements should appear after the **sync_group** statement and before **wideip** statements.

```
server {
  type 3dns
  address <IP address>
  name <"3dns_name">
  iquery_protocol [ udp | tcp ]
  [ remote {
    secure <yes | no>
    user <"user name">
    } ]
  [ interface {
    address <NIC IP address>
    address <NIC IP address>
    } ]
  [ factories {
    prober <number>
    discovery <number>
    snmp <number>
    hops <number>
    } ]
  [ prober <IP address> ]
  probe_protocol < icmp | udp | tcp | dns_ver| dns_dot>
  port <port to probe>
}
```

Figure 4.4 *Syntax for defining a 3-DNS Controller server*

Figure 4.5 shows a sample **server** statement that defines a 3-DNS Controller.

```
// New York
server {
   type 3dns
   address 192.168.101.2
   name "3dns-newyork"
   iquery_protocol  udp
   remote {
      secure no
      user "root"
   }
   prober 192.168.101.40
   probe_protocol icmp
   port 53
}
```

Figure 4.5 *Sample 3-DNS Controller server definition*

Defining BIG-IP Controller servers

Before you define BIG-IP Controller servers, you should have the following information:

- The IP address and service name or port number of each virtual server to be managed by the BIG-IP Controller
- The IP address of the server itself

To define a BIG-IP Controller server using the Configuration utility

1. In the navigation pane, click **BIG-IP Controllers**.
2. On the toolbar, click **Add BIG-IP Controller**.
   The Add New BIG-IP Controller screen opens.

3. Add the new BIG-IP Controller settings. (For help on defining BIG-IP Controllers, click **Help** on the toolbar.) The BIG-IP Controller and specified virtual server are added to your configuration.

To add more virtual servers using the Configuration utility

1. In the navigation pane, click **BIG-IP Controllers**.
2. In the table, find the BIG-IP Controller that you just added.
3. Click the entry in its **BIG-IP Virtual Servers** column.
4. On the toolbar, click **Add Virtual Server**.
   The Add Virtual Server to BIG-IP screen opens.
5. Add the new virtual server settings. For help on adding virtual servers, click **Help** on the toolbar.

Repeat this process for each virtual server you want to add to this BIG-IP Controller.

To define a BIG-IP Controller server manually

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Use the syntax shown in Figure 4.6 to define a BIG-IP Controller.

All **server** statements should appear after the **sync_group** statement and before **wideip** statements.

If you need to allow iQuery packets to pass through firewalls, include the **translate** keyword in the **server** statement that defines the BIG-IP Controller. When you include the **translate** keyword, the iQuery utility includes translated IP addresses in the packets sent to the specific BIG-IP Controller. See *Setting up iQuery communications for the big3d agent*, on page 2-21 for details.

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

  vs {
    address <virtual server IP address>
    port <port number> | service <"service name">
    [ translate {
        address <IP address>
        port <port number>|service <"service name">
      } ]
   }
}
```

Figure 4.6 *Syntax for defining a BIG-IP Controller server*

Figure 4.7 shows a sample **server** statement that defines a BIG-IP Controller.

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

Figure 4.7 *Sample BIG-IP Controller server definition*

Defining host servers

A host is an individual network server or server array controller other than the BIG-IP Controller. Before configuring a host, you should have the following information:

- **Address information**
  The IP address and service name or port number of each virtual server to be managed by the host.
- **SNMP information for host probing**
  To implement host probing, you must specify SNMP agent settings after you define the host server. The settings you specify include the type and version of SNMP agent that runs on the host, the community string, and the number of communication attempts that you want the **big3d** agent to make while gathering host metrics. SNMP agent settings for hosts are described in *Configuring host SNMP settings*, on page 4-15.

Note

*To fully configure host probing, you must configure the SNMP agent settings in the host definition as previously described, and you must also set up the **big3d** agents to run SNMP factories, and configure the SNMP agents on the hosts themselves. See Setting up SNMP probing for hosts, on page 2-14* for details.

To define a host server using the Configuration utility

1. In the navigation pane, click **Host Servers**.
2. On the toolbar, click **Add Host Server**.
   The Add New Host Server screen opens
3. Add the new host server settings. For help on adding host servers, click **Help** on the toolbar.
   The host and the specified virtual server are added to your configuration.

To add more virtual servers using the Configuration utility

1. In the navigation pane, click **Host Servers**.
2. In the table, find the host that you just added.
3. Click the entry in its **Host Virtual Servers** column.
4. On the toolbar, click **Add Host Virtual Server**. The Add Virtual Server to Host screen opens.
5. Add the new virtual server settings. For help on adding virtual servers, click **Help** on the toolbar.

Repeat this process for each virtual server you want to add to this host.

To define a host server manually

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Use the syntax shown in Figure 4.8 to define a host.

   All **server** statements should appear after the **sync_group** statement and before **wideip** statements.

```
server {
  type host
  address <IP address>
  name <"host_name">
  [ prober <ip_address> ]
  probe_protocol <tcp | icmp | udp | dns_ver | dns_dot>
  port <port number> | service <"service name">
  [ snmp {
    agent <generic | ucd | solstice | ntserv | ciscoldv2 |
ciscoldv3>
    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 | udp | dns_ver| dns_dot> ]
  }
}
```

Figure 4.8 *Syntax for defining a host server*

Figure 4.9 shows a sample **server** statement that defines a host.

```
server {
   type            host
   address         192.168.104.40
   name            "host-tokyo"
   prober          192.168.101.40
   probe_protocol  icmp
   port            53
   snmp {
      agent        ucd
      community    "public"
      version      1
   }
   vs {
      address         192.168.104.50:25
   }
   vs {
      address         192.168.104.50:80
   }
}
```

Figure 4.9 *Sample host server definition*

Configuring host SNMP settings

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 or BIG-IP Controller that runs the **big3d** agent.

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

- Memory utilization
- CPU utilization
- Disk space utilization
- Packet rate

The 3-DNS Controller supports the following host SNMP agents:

- **Generic**
  A generic SNMP agent is an SNMP agent that collects metrics provided by OIDs as specified in the RFC 1213 document.
- **UCD SNMPD**
  This free SNMP agent is provided by the University of California at Davis. It is available on the web at **http://ucd-snmp.ucdavis.edu,** or you can download the **ucd-snmp.tar.gz** file from **ftp://ucd-snmp.ucdavis.edu**.
- **Solstice Enterprise**
  This SNMP agent is a product of Sun Microsystems.
- **Windows NT 4.0 SNMP**
  This SNMP matrix agent is a product of Microsoft and is distributed with the Microsoft Windows NT 4.0 server.
- **Cisco LDV2**
  This SNMP agent is a product of Cisco and is distributed with the Cisco LocalDirector, version 2.X.
- **Cisco LDV3**
  This SNMP agent is a product of Cisco and is distributed with the Cisco LocalDirector, version 3.X.

Configuring SNMP agents on hosts, on page 4-18, provides some useful tips for configuring the different SNMP agents on the hosts themselves. We recommend that you use the information in conjunction with the documentation originally provided with the SNMP agent.

To configure host SNMP settings using the Configuration utility

1. In the navigation pane, 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 manually

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate 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. Figure 4.10 shows the SNMP syntax in bold.
6. Define the virtual server information as usual.

```
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 | ntserv | 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 4.10 *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. The following sections offer some tips and hints on configuring each type of supported SNMP agent, but you may want to refer to the documentation provided with your SNMP software for more complete configuration information.

Configuring the UCD SNMP agent on the host

The UCD SNMP agent runs on HP-UX, Ultrix, Solaris, SunOS, OSF, NetBSD, FreeBSD, BSDi, Linux, AIX, OpenBSD, Irix, Windows 95, and Windows NT. Please refer to the **ucdFAQ.txt** file for details. On UNIX and UNIX-like systems, the default

location for the configuration and MIB files is in the **/usr/share/snmp** directory. You can find help on **snmpd** options in the snmpd man page.

Figure 4.11 shows a sample configuration file in **/usr/share/snmp/snmpd.conf**. This file configures the SNMP agent to define a community. Our example uses **3dnspwd** as the community, which is retrieved from the address **192.168.254.4** using the prober at **192.168.254.240**. It allows read access of the entire SNMP MIB tree, but does not allow write access.

```
------------begin /usr/share/snmp/snmpd.conf------------
#
# To allow write access to the 'system' subgroup from the local
# network with the community string "sysadmin":
#
# - amend the "source" address in the com2sec section
# to match your local network address
# - uncomment the "access admin" line below
#
# You are also strongly advised to change the community string
# to something other than "sysadmin"
# sec.name source community
com2sec local localhost private
com2sec 3dns 192.168.254.240/32 3dnspwd
# sec.model sec.name
group local any local
group public any public
group 3dnsgroup any 3dns
# incl/excl subtree mask
view all included .1 80
view system included system fe
view mib2 included .iso.org.dod.internet.mgmt.mib-2 fc
# context sec.model sec.level prefix read write not
#access admin "" any noauth 0 mib2 system none
access public "" any noauth 0 system none none
access local "" any noauth 0 all all all
access 3dnsgroup "" any noauth 0 all none none
------------eof /usr/share/snmp/snmpd.conf------------
```

Figure 4.11 *Configuring a UCD SNMP agent on the host*

Figure 4.12 shows the corresponding host **server** statement.

```
server {
    type host
    address 192.168.254.4 # address of host + SNMP agent
    prober 192.168.254.240 # SNMP prober reader
    snmp {
       agent ucd
       community 3dnspwd
    }
    vs {
       address 192.168.254.201
    }
    : : :
}
```

Figure 4.12 *Configuring the host server statement to run the UCD SNMP agent*

Configuring the Solstice SNMP agent on the host

The Solaris or SunOS 5.x should include the Solstice Master Agent in the distribution CD. Figure 4.13 shows a sample configuration that should work for host probing.

```
-------begin /etc/snmp/conf/snmpd.conf------
# Copyright 1988-01/28/97 Sun Microsystems, Inc. All Rights Reserved.
#pragma ident "@(#)snmpd.conf 2.22 97/01/28 Sun Microsystems"
# See below for file format and supported keywords
sysdescr Sun SNMP Agent,
syscontact System administrator
sysLocation System administrators office
#
system-group-read-community public
#system-group-write-community private
#
read-community public
#write-community private
#
trap localhost
trap-community SNMP-trap
#
#kernel-file /vmunix
#
#managers 192.168.254.240
##############################
# File Format:
# Each entry consists of a keyword followed by a parameter
# string, terminated by a newline. The keyword must begin in the
# first position. The parameters are separated from the keyword
# (and fromone another) by whitespace. All text following (and
# including) a '#' character is ignored. Case in keywords is
# ignored, but case inparameter strings is NOT ignored.
```

Figure 4.13 *Configuring a Solstice SNMP agent on the host (continued on next page)*

```
# Supported Keywords:
# sysdescr String to use for sysDescr.
# syscontact String to use for sysContact.
# syslocation String to use for sysLocation.
# system-group-read-community Community name needed for read
# access to the system group.
# system-group-write-community Community name needed for write
# accessto the system group.
# read-community Community name needed for read access
# to the entire MIB.
# write-community Community name needed for write access
# to the entire MIB (implies read access).
#
# trap Host names where traps should be sent.
# A maximum of 5 hosts may be listed.
# trap-community Community name to be used in traps.
#
# kernel-file Filename to use for kernel symbols.
#
# managers Hosts that can send SNMP queries.
# Only five hosts may be listed on any one line.
# This keyword may be repeated for a total of 32 hosts.
#
# newdevice Additional devices which are not built in snmpd
# format as below
#
# newdevice type speed name
#
# where newdevice is keyword, type is an interger which has to
# match yourschema file, speed is the new device's speed, and
# name is this newdevice's name
------eof /etc/snmp/conf/snmpd.conf------
```

Figure 4.13 *Configuring a Solstice SNMP agent on the host (continued from previous page)*

This allows **192.168.254.240** to query the Solstice SNMP agent—and its community is **public**. The **wideip.conf** would be similar to the example for UCD except that the community is **"public."**

Configuring the Windows NT 4.0 SNMP agent on the host

To configure the Windows NT 4.0 SNMP agent, you need to complete five tasks:

- Install the SNMP agent
- Configure the SNMP server
- Install the Windows NT resource kit
- Verify that the server is running
- Verify that the installation is good

To install the SNMP agent via the Network Services

1. Right-click the Network Neighborhood icon on your desktop.
2. From the popup menu, select **Properties**.
3. In the Properties dialog box, click the Services tab.
4. Click **Add**, and then choose the SNMP service from the service list.
5. Configure the community name, IP address allowed to query, and so on, to reflect the same configuration as specified in the **wideip.conf** file.

For the SNMP agent to work, you must reinstall, into your Windows NT server, whatever service pack you have previously installed.

To configure the SNMP server

When you configure the SNMP server, you need to provide the contact, community, and permission information that allows the **big3d** agent to read the SNMP MIB. You cannot change the SNMP configuration when the SNMP service is running. However, you can temporarily stop the SNMP service by typing **net stop snmp** at the command prompt. Then make the configuration changes, and when you are finished, restart the service by typing **net start snmp**.

To install the Windows NT Resource Kit

If you are doing a typical setup, you should install the Windows NT Resource Kit (if it is not already installed on the server). These utilities should provide you with the following important files:

- **MIBCC.EXE** (MIB compiler)
- **SNMPMON.EXE** (SNMP monitor)
- **SNMPUTIL.EXE** (**get/walk/getnext** utility)
- **PERF2MIB.EXE**
- **LMMIB2.MIB**
- **MIB_II.MIB**
- **SMI.MIB**

To verify that the SNMP server is running

1. Click the Services tab and make sure the SNMP server is up and running.
2. From the directory where you installed the resource kit utilities, run the following at the command prompt:

   ```
   c:\utilities\perfm
   ```

   The **perfm.bat** file effectively creates the performance monitoring agent's **.dll**, automatically loads it, and then restarts the SNMP agent.

To verify the installation

To verify that the Windows NT SNMP is working, use the 3-DNS Controller or BIG-IP Controller that runs the **big3d** SNMP factory. Run either the **snmptest** or **snmpwalk** commands.

Note

*Before running **snmptest** or **snmpwalk,** be sure that the ephemeral ports are open by typing the command:*
sysctl -w bigip.open_3dns_lockdown_ports=1

WARNING

We strongly recommend that you do not run a screensaver on your Windows NT server when it is running an SNMP agent. If you run a screensaver and the SNMP agent simultaneously, the CPU utilization reported by NT may show as 100% busy.

Configuring the Cisco SNMP agent on the host

The Cisco LocalDirector versions 2.x and 3.x should include the Cisco SNMP agent in the distribution CD. Figure 4.14 is a sample configuration (in the **ciscold.txt** file) that should work for host probing.

```
: Saved
: LocalDirector 410 Version 3.1.3
syslog output 5.5
no syslog console
enable password c88f22962f5d2b7e09cc8fbf48f92b encrypted
hostname localdirector
no shutdown ethernet 0
no shutdown ethernet 1
shutdown ethernet 2
interface ethernet 0 auto
interface ethernet 1 auto
interface ethernet 2 auto
mtu 0 1500
mtu 1 1500
mtu 2 1500
multiring all
no secure 0
no secure 1
no secure 2
ping-allow 0
ping-allow 1
no ping-allow 2
ip address 192.168.254.6 255.255.255.0
no rip passive
failover ip address 192.168.254.7
no failover
password foobar
```

Figure 4.14 *Configuring a Cisco SNMP agent on the host (continued on next page)*

```
telnet 192.168.254.0 255.255.255.0
snmp-server host 192.168.254.206
snmp-server host 192.168.254.4
snmp-server host 192.168.254.238
snmp-server host 192.168.254.240
snmp-server enable traps
snmp-server contact SystemAdministration's name
snmp-server location F5 5/F SystemAdmin's location
tftp-server 192.168.254.206 port 69 /usr/sysadm/f5/ciscold
virtual 192.168.254.201:80:0:tcp is
virtual 192.168.254.202:80:0:tcp is
virtual 192.168.254.203:80:0:tcp is
predictor 192.168.254.201:80:0:tcp roundrobin
predictor 192.168.254.202:80:0:tcp roundrobin
predictor 192.168.254.203:80:0:tcp roundrobin
real 192.168.254.10:80:0:tcp is
real 192.168.254.11:80:0:tcp is
real 192.168.254.12:80:0:tcp is
real 192.168.254.13:80:0:tcp is
real 192.168.254.14:80:0:tcp is
no names
bind 192.168.254.201:80:0:tcp 192.168.254.10:80:0:tcp
192.168.254.11:80:0:tcp
bind 192.168.254.202:80:0:tcp 192.168.254.12:80:0:tcp
192.168.254.13:80:0:tcp
bind 192.168.254.203:80:0:tcp 192.168.254.14:80:0:tcp
: end
```

Figure 4.14 *Configuring a Cisco SNMP agent on the host (continued from previous page)*

Figure 4.15 shows the corresponding host **server** statement.

```
server {
    type host
    address 192.168.254.6
    vsmetrics yes
    snmp {
       agent ciscold3
    }
      vs {
        address 192.168.254.201:80
    }
}
    : : : :
```

Figure 4.15 *Configuring the host server statement to run the ciscold3 SNMP agent*

Setting up sync groups

A sync group defines the group of 3-DNS Controllers that synchronize their configuration settings and metrics data. You configure a sync group from the principal 3-DNS Controller. First list the IP address of the principal itself. Then list all other 3-DNS Controllers in the order that they should become principals if previously listed 3-DNS Controllers fail.

Each 3-DNS Controller in your network must be included in a sync group. There may be cases where you do not want a 3-DNS Controller to share its configuration with other controllers. In this case, you can create a separate sync group for each 3-DNS Controller. Each sync group would contain only its own name or IP address.

```
sync_group {
  name "sync-ny"
  3dns 192.168.101.2    // New York
}

sync_group {
  name "sync-la"
  3dns 192.168.102.2    // Los Angeles
}
```

Figure 4.16 *Sample non-syncing* ***sync groups*** *statements*

◆ Note

To implement such a configuration, you must modify each 3-DNS Controller's ***wideip.conf*** *file; the Configuration utility does not support this function.*

To define a sync group using the Configuration utility

1. In the navigation pane, click **3-DNS Sync**.
   The System - Add a New Sync Group screen opens.
2. In the **New Sync Group Name** box, type the name of the new sync group and click **Add.**
   The Add a 3-DNS to a Sync Group screen opens.
3. From the list of 3-DNS Controllers, first select the 3-DNS Controller that you want to be the principal controller. Then check the box next to each 3-DNS Controller that you want to add to the sync group.
4. Click **Add**.

To define a sync group manually

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Use the syntax shown in Figure 4.17 to define sync groups.

   The **sync_group** statement should appear after the **datacenter** statement and before **server** statements.

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

Figure 4.17 *Syntax for setting up a sync group*

Figure 4.18 shows a sample **sync_group** statement.

```
sync_group {
  name "sync"
  3dns 192.168.101.2    // New York
  3dns 192.168.102.2    // Los Angeles
}
```

Figure 4.18 *Sample sync group definition*

Setting the time tolerance value

The time tolerance value is a global variable that defines 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. See *Understanding how the time tolerance variable affects sync groups*, on page 2-9 for details.

To check the value for the time tolerance setting using the Configuration utility

1. In the navigation pane, click **System**.
   The System - General screen opens.
2. On the toolbar, click **Timers and Task Intervals**.
3. Note the value in the **3-DNS Sync Time Tolerance** box, and change it if necessary.
4. If you change this setting, click **Update** to save it.

To check the value for the time tolerance setting in the configuration file

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Search for **time_tolerance**. If the **time_tolerance** sub-statement is not in the configuration file, the default (**10**) is used.

Configuring global variables

Default values for global parameters are adequate for most situations. However, we recommend that you specifically enable encryption for crypto 3-DNS Controllers.

To configure global parameters using the Configuration utility

1. In the navigation pane, click **System**.
   The System - General screen opens. Note that global parameters are grouped into several categories on this screen. Each category has its own toolbar item, and online help is available for each parameter.
2. Make general global changes at the System - General screen or, to make changes to global parameters in other categories, click the appropriate toolbar item.
3. Add the new global settings. For help on configuring the global settings, click **Help** on the toolbar.

The new global parameters are added to your configuration.

To configure global parameters manually

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

   For example, to enable encryption for iQuery transactions (which is recommended), change the encryption parameter to **yes** (the default setting is **no**). If you want to use a non-default name for the encryption key file, type it on the next line.

Figure 4.19 shows the correct syntax for enabling encryption.

```
globals {
   encryption yes
   encryption_key_file "/etc/F5key.dat"
}
```

Figure 4.19 *Syntax for enabling encryption*

For descriptions of all global parameters, see *The globals statement*, on page A-7.

Configuring IP filters

Filters control network traffic by specifying whether packets are accepted or rejected by the 3-DNS Controller. Filters apply to both incoming and outgoing traffic. When creating a filter, you define the criteria to apply to each packet that is processed by the 3-DNS Controller. You can configure the 3-DNS Controller to accept or block each packet based on whether the packet matches the criteria.

Typical criteria that you define in IP filters are packet source IP addresses, packet destination IP addresses, and upper-layer protocol of the packet. However, each protocol has its own specific set of criteria that can be defined.

For a single filter, you can define multiple criteria in multiple, separate statements. To tie the statements to the same filter, each of these statements should reference the same identifying name or number. You can have as many criteria statements as you want, limited only by the available memory. Of course, the more statements you have, the more difficult it is to understand and maintain your filters.

Defining the filter criteria

When you define an IP filter, you can filter traffic in two ways:

- You can filter traffic going to a specific destination or coming from a specific destination, or both.
- The filter can allow network traffic through, or it can reject network traffic.

To define an IP filter using the Configuration utility

1. In the navigation pane, click **IP Filters**.
   The IP Filters screen opens.
2. On the toolbar, click **Add Filter**.
   The Add IP Filter screen opens
3. Add the IP filter settings. For help on configuring the IP filter, click **Help** on the toolbar.

Note

For information on configuring IP filters and rate filter on the command line, refer to the IPFW man page.

Configuring Sendmail

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

Before you begin setting up Sendmail, you may need to look up the name of the mail exchanger for your domain. If you already know the name of the mail exchanger, refer to *Setting up Sendmail*, on page 4-37 for details about setting up the **sendmail** daemon itself.

Finding the mail exchanger for your domain

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

To find the mail exchanger

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

   ```
   /etc# nslookup
   ```

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

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

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

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

The returned information includes the name of the mail exchanger. For example, the sample information shown in Figure 4.20 lists **bigip.net** as the preferred mail exchanger.

```
bigip.net   preference = 10, mail exchanger = mail.SiteOne.com
bigip.net   nameserver = ns1.bigip.net
bigip.net   nameserver = ns2.bigip.net
bigip.net   internet address = 192.17.112.1
ns1.bigip.net       internet address = 192.17.112.2
ns2.bigip.net       internet address = 192.17.112.3
```

Figure 4.20 *Sample mail exchanger information*

Setting up Sendmail

When you set up Sendmail, you must edit a couple of configuration files. Since the 3-DNS Controller does not accept email messages, you can use the **crontab** utility to purge unsent or returned messages and send them to yourself or another administrator.

To set up and start Sendmail

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

   ```
   DS<MAILHUB_OR_RELAY>
   ```

3. Save and close the **/etc/sendmail.cf** file.
4. To allow Sendmail to flush outgoing messages from the queue containing mail that cannot be delivered immediately, open the **/etc/crontab** file, and change the last line of the file to read:

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

5. Save and close the **/etc/crontab** file.
6. To prevent returned or undelivered email from going unnoticed, open the **/etc/aliases** file and create an entry so **root** points to you or another administrator at your site.

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

7. Save and close the **/etc/aliases** file.
8. Run the **newaliases** command to generate a new aliases database that incorporates the information you added to the **/etc/aliases** file.

9. To turn Sendmail on, either reboot the system or type the following command:

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

◆ Note

The 3-DNS Controller supports only outgoing mail for Sendmail servers.

F5 Networks
3·DNS®
Distributed Traffic Controller

5

Configuring Basic Load Balancing

- Getting started with load balancing
- Configuring load balancing
- Changing global variables that affect load balancing

Getting started with load balancing

The third and final phase of installing 3-DNS Controllers is to configure load balancing modes. The 3-DNS Controllers use these load balancing modes when resolving DNS name resolution requests sent by LDNS servers.

This chapter first describes the various load balancing modes, and later 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 (containing one or more IP addresses), 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 basic load balancing mode, which selects a server based on a pre-defined pattern, or an advanced 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, and 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.
- **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 this 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 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
Static Persist		x	x	x
Topology	x	x	x	x
VS Capacity		x	x	x

Table 5.1 *Load balancing mode usage*

Using basic, static load balancing modes

Basic 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 basic 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 remaining basic 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. 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.

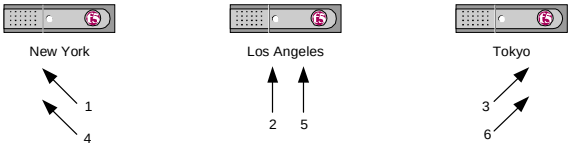


Figure 5.1 *Round Robin mode*

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.

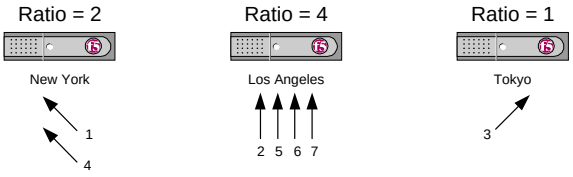


Figure 5.2 *Ratio mode*

Random mode

Random mode sends connections to virtual servers in a random pattern.

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 the configuration file. This allows you to develop proximity-based mapping. For example, customers in a particular geographic region can be sent to servers within that same region. The 3-DNS Controller determines the proximity of servers by comparing the client's LDNS IP address to the IP address of the available servers.

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 Topology load balancing mode is different from the topology-based access control feature. Topology-based access control actually prevents clients from connecting to specific virtual servers. You can use the topology-based access control feature in conjunction with the Topology load balancing mode. See Chapter 6, *Configuring Specialized Load Balancing*, for detailed information about working with this and other topology features.

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. 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 advanced, dynamic load balancing modes

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

All advanced 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 can define when you set the global TTL (time to live) variables.

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

- Completion Rate
- Least Connections
- Packet Rate
- Round Trip Times (RTT)

- Hops
- Quality of Service
- 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.

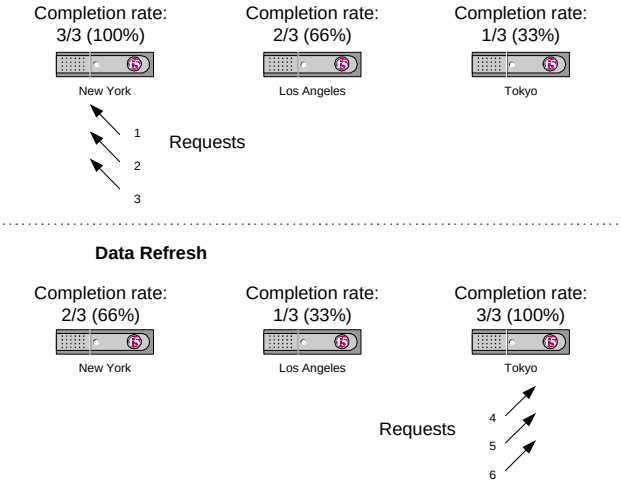


Figure 5.3 *Completion Rate mode*

Least Connections mode

Least Connections mode is also 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.

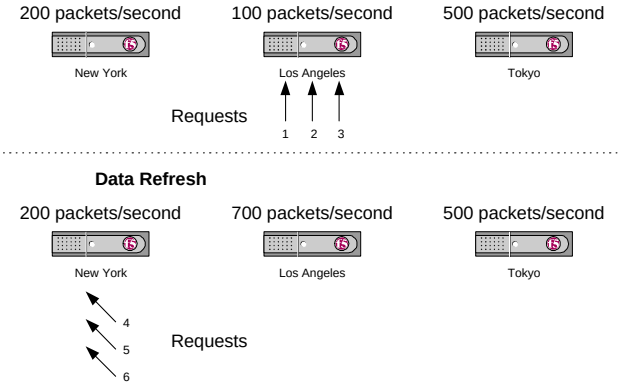


Figure 5.4 *Packet Rate mode*

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. This load balancing mode requires that you run one or more **big3d** agents in each data center to collect the required metrics.

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

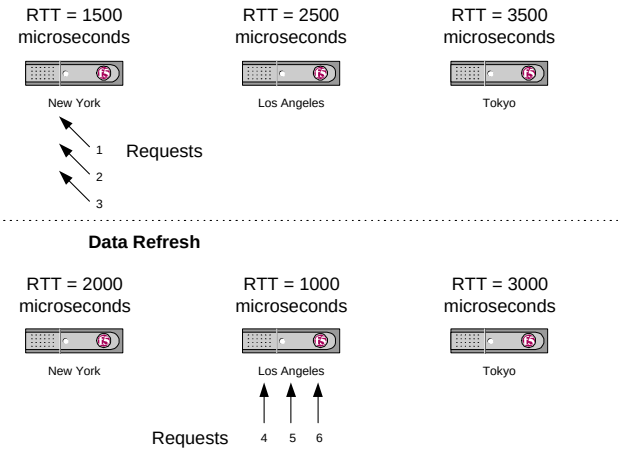


Figure 5.5 *Round Trip Times mode*

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.

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

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 Chapter 6, *Configuring Specialized Load Balancing*.

VS Capacity mode

VS Capacity mode selects the virtual server which has the most nodes **up**.

Configuring load balancing

This section describes how to configure load balancing. For information on how to implement the more specialized load balancing modes, see Chapter 6, *Configuring Specialized Load Balancing*.

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 on page 5-17.
- **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 following section, *Adding a wide IP*.

Understanding wide IPs

After you configure the BIG-IP Controllers, 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 to a set of virtual servers that host the domain content, such as a web site or an e-commerce site.

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

- Gather your BIG-IP Controller and host configuration information so you can easily see which virtual servers have the replicated content. Then you can decide how to group virtual servers into pools.
- Decide which load balancing modes to use for each pool of virtual servers. If you need to review the available load balancing modes, see *Choosing a load balancing mode*, on page 2-27.

◆ 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. 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). If you want to manually configure the 3-DNS Controller, see ***Relating BIND information to 3-DNS Controller wide IP definitions****, on page C-10.*

There may be situations (for example, e-commerce and FTP sites) 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 the wide IP is defined. For details, see *Setting up load balancing for services that require multiple ports*, on page 6-11.

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 both types of virtual servers (BIG-IP Controller 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**.
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 manually define a wide IP

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. 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's address, port, and name information. Enclose the wide IP's 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 5.6 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>
    } ]
  [ pool_lbmode <rr | ratio | ga | random | topology> ]
  pool {
    name <"pool_name">
    [ ratio <pool_ratio> ]
    [ dynamic_ratio < yes | no > ]
    [ rr_ldns < yes | no > ]
    [ preferred < completion_rate | ga | hops | 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 5.6 *Syntax for the wideip statement*

Figure 5.7 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 randomly. The 3-DNS Controller distributes requests to the two defined virtual servers in **pool_2**, at a 2:1 ratio 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 5.7 *Example syntax for defining a wide IP*

Troubleshooting manual configuration problems

Adding a wide IP requires careful planning and use of correct syntax. We have included the following recommendations to make it easier for you to spot and resolve any configuration problems:

- **Configuration utility**
  The Configuration utility contains Statistics screens that are useful in diagnosing problems, as they provide a snapshot of your 3-DNS Controller network at any given time. To use them, click the Expand button [+] to the left of the **Statistics** item in the navigation pane, then click either **Wide IPs** or **Summary** (and scroll until you see the **Wide IP** table).
- **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. For an example of a **wideip.conf** file, see Appendix A, *Wideip.conf Syntax*.
- **/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:
  - Appendix C of this manual
  - The O'Reilly & Associates book, ***DNS and BIND***, 3rd 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. 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 or to *The globals statement*, on page A-7.

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, select **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 5.8 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 5.8 *Configuring global alternate and fallback load balancing modes*

Figure 5.9 shows a sample **globals** statement that defines global load balancing variables.

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

Figure 5.9 *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. After a timeout is reached, the 3-DNS Controller refreshes the information.

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

Parameter	Description	Default
BIG-IP TTL	Specifies the number of seconds that the 3-DNS Controller will use BIG-IP Controller metrics information for name resolution and load balancing.	60
Host TTL	Specifies the number of seconds that the 3-DNS Controller will use 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 will use virtual server information (data acquired from a BIG-IP Controller 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 will use 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 **A** record to be valid. If you do not specify a wide IP TTL value when defining a wide IP, the wide IP definition uses the **default_ttl** value.	30

Table 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
BIG-IP data refresh	Specifies the frequency (in seconds) at which the 3-DNS Controller refreshes BIG-IP Controller 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
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	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. Note: If you are using NTP to synchronize the time of the 3-DNS Controller with a time server, select a time tolerance of **0**.	10
Timer Sync State	Specifies the interval (in seconds) at which the 3-DNS Controller checks to see if it should change states (from principal to receiver or from receiver to principal). 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.	30
Persist Cache	Specifies the interval (in seconds) at which the 3-DNS Controller discards the paths and other metrics data.	300

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, select **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 TTL and timer values.

```
globals {
  [ timer_get_3dns_data <number> ]
  [ timer_get_bigip_data <number> ]
  [ timer_get_host_data <number> ]
  [ timer_get_vs_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> ]
  [ bigip_ttl <number> ]
  [ host_ttl <number> ]
  [ vs_ttl <number> ]
  [ path_ttl <number> ]
  [ trace_ttl <number> ]
  [ default_ttl <number> ]
}
```

Figure 5.10 *Syntax for configuring global TTL and timer values*

F5 Networks
3·DNS®
Distributed Traffic Controller

6

Configuring Specialized Load Balancing

- Configuring load balancing using specialized modes
- Setting up Quality of Service (QOS) mode
- Setting up Global Availability mode
- Setting up load balancing for services that require multiple ports
- Setting up topology-based features

Configuring load balancing using specialized modes

This chapter describes the following specialized load balancing modes:

- Quality of service
- Global availability
- E-commerce
- Topology access control
- Topology load balancing

You configure a specialized load balancing mode just as you do the standard modes, by editing the wide IP definition. See *Adding a wide IP*, on page 5-12. You can also set global alternate and fallback load balancing modes. See *Setting global alternate and fallback modes*, on page 5-17.

Setting up Quality of Service (QOS) mode

The Quality of Service (QOS) mode is a user-definable mode that includes a configurable combination of the RTT, Completion Rate, Packet Rate, Topology, Hops, and VS Capacity modes. The QOS mode is based on an equation that takes each of these performance factors into account. When the 3-DNS Controller selects a virtual server, it chooses the server with the best overall score.

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

Understanding QOS coefficients

The following table lists each QOS coefficient, its scale, a likely upper limit for each, and whether a higher or lower value is more efficient.

Coefficient	How measured	Example upper limit	Higher or lower?
Packet rate	Packets per second	700	Lower
Round trip time	Microseconds	2,000,000	Lower
Completion rate	Percentage of successfully transferred packets (0-100%)	100%	Higher
Topology	Score that defines network proximity by comparing server and LDNS IP addresses ($0\text{-}2^{32}$)	100	Higher
Hops	Number of intermediate systems transitions (hops)	64	Lower
VS capacity	Number of nodes ***up***	20	Higher

Table 6.1 *QOS coefficients: ranges and limits*

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

- **Scale**
  The raw metrics for each coefficient are not on the same scale. For example, completion rate is measured in percentages while the packet rate is measured in packets per second.
- **Normalization**
  The 3-DNS Controller normalizes the raw metrics to values in the range of 0 to10. As the QOS value is calculated, a high measurement for completion rate is good, because a high percentage of completed connections are being made, but a high value for packet rate is not desirable because the packet rate load balancing mode attempts to find a virtual server that is not overly taxed at the moment.

- **Emphasis**
  You can adjust coefficients to emphasize one normalized metric over another. For example, by changing the coefficients to the values shown in Figure 6.1, you are putting the most emphasis on completion rate.

```
globals {
         qos_coeff_rtt 20
         qos_coeff_completion_rate 100
         qos_coeff_packet_rate 50
         qos_coeff_topology 0
         qos_coeff_hops 0
         qos_coeff_vs_capacity 0
}
```

Figure 6.1 *Emphasizing completion rate*

In the preceding example, if completion rates for two virtual servers are close, the virtual server with the best packet rate is chosen. If both completion rates and packet rates are close, the round trip time (RTT) breaks the tie. In this example, the metrics for topology, hops, and vs modes are not used in determining how to distribute connections.

Customizing the QOS equation

You can customize the QOS equation globally, meaning that the equation applies to all wide IPs that use the QOS mode. You can also customize individual wide IPs, in which case the global QOS equation settings are overwritten.

To assign global QOS coefficients using the Configuration utility

1. In the navigation pane, click **System**.
   The System - General screen opens.
2. On the toolbar, click **Load Balancing**.
   The System - Load Balancing screen opens.
3. Define the global QOS coefficients in the **Round Trip Time**, **Completion Rate**, **Hops**, **BIG-IP Packet Rate**, **Topology,** and **VS Capacity** boxes.
4. Click **Update**.

To assign QOS coefficients for a specific wide IP using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
2. In the Wide IP column, click a wide IP name.
   The Modify Wide IP screen opens.
3. On the toolbar, click **Modify Pool**.
   The Modify Wide IP Pools screen opens.
4. In the Pool Name column, click the name of a pool.
   The Modify Load Balancing screen opens.
5. Define the wide IP's QOS coefficients in the **Round Trip Time**, **Completion Rate**, **Hops**, **Packet Rate**, **Topology,** and **VS Capacity** boxes.
6. Click **Update**.

To manually assign global QOS coefficients

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

4. Refer to the example syntax shown in Figure 6.2 to define a global QOS equation.

```
globals {
         qos_coeff_rtt 20
         qos_coeff_completion_rate 5
         qos_coeff_packet_rate 3
         qos_coeff_topology 0
         qos_coeff_hops 0
         qos_coeff_vs_capacity 0
}
```

Figure 6.2 *Sample global QOS equation*

To manually assign QOS coefficients for a specific wide IP

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement you want to edit.
4. Refer to the example syntax shown in Figure 6.3 to define a wide IP's QOS equation. Figure 6.3 displays a wide IP definition that uses overrides for the global settings shown in Figure 6.2.

```
wideip {
   address           192.168.101.50
   service           "http"
   name              "www.wip.domain.com"
   ttl               60      // increase the domain default ttl
   qos_coeff {
      rtt             21
      hops            0
      completion_rate 7
      packet_rate     5
      topology        1
      vs_capacity     0
   }
   pool {
      name           "Pool_1"
      ratio          2         // applies to pool_lbmode == ratio
      preferred      qos
      alternate      ratio
      address       192.168.101.50   ratio 2
      address       192.168.102.50   ratio 1
      address       192.168.103.50   ratio 1
   }
   pool {
      name           "Pool_2"
      ratio          1
      preferred      rr
      address       192.168.102.60   ratio 2
      address       192.168.103.60   ratio 1
   }
}
```

Figure 6.3 *QOS coefficient settings that override the global default settings*

Using the Dynamic Ratio option

When the Dynamic Ratio option is turned on, the 3-DNS Controller treats QOS scores as ratios, and it uses each server in proportion to the ratio determined by the QOS calculation. When the Dynamic Ratio option is turned off (the default), the 3-DNS Controller uses

only the server with the highest QOS score for load balancing (in which case it is a winner takes all situation) until metrics information is refreshed.

To turn on the Dynamic Ratio option using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
2. In the Wide IP column, click a wide IP name.
   The Modify Wide IP screen opens.
3. On the toolbar, click **Modify Pool**.
   The Modify Wide IP Pools screen opens.
4. In the Pool Name column, click the name of a pool.
   The Modify Load Balancing screen opens.
5. Check **Use Dynamic Ratio**.
6. Click **Update**.

To manually turn on the Dynamic Ratio option

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement and the pool definition you want to edit.
4. Add the syntax (shown in bold in Figure 6.4) to the pool definition.

```
pool {
  name <"pool_name">
  [ ratio <pool_ratio> ]
  dynamic_ratio yes
  [ rr_ldns < yes | no > ]
  [ rr_ldns_limit <number> ]
  [ preferred < completion_rate | ga | hops | leastconn |
    packet_rate | qos | random | ratio | return_to_dns | rr |
    rtt | topology | 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 6.4 *Enabling dynamic ratio*

Setting up Global Availability mode

The global availability mode selects the first available virtual server in a wide IP definition. If that virtual server becomes unavailable, subsequent connections go to the next listed virtual server in the wide IP definition.

The 3-DNS Controller always starts with the first virtual server in the list. Over time, the first server in the list receives the most connections, and the last server in the list receives the fewest connections. Figure 6.5 shows the 3-DNS Controller using the global availability load balancing mode.

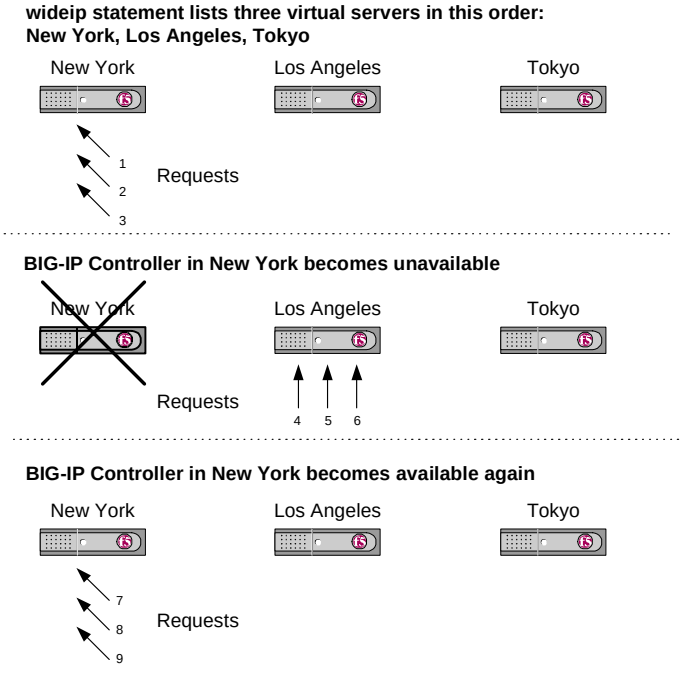


Figure 6.5 *Global Availability mode*

To implement the global availability load balancing mode using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
2. In the Wide IP column, click a wide IP name.
   The Modify Wide IP screen opens.
3. On the toolbar, click **Modify Pool**.
   The Modify Wide IP Pools screen opens.
4. In the Pool Name column, click the name of a pool.
   The Modify Load Balancing screen opens.
5. Select **Global Availability** as the **Preferred**, **Alternate**, or **Fallback** load balancing mode.
6. Click **Update**.

7. A popup screen appears, indicating that with the Global Availability load balancing mode you must order the virtual servers. Click **OK**.
   The Modify Virtual Servers screen opens.
8. In the Order column, specify the order in which you want to list the virtual servers for Global Availability.
9. Click **Update**.

To manually implement the global availability load balancing mode

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement you want to edit.
4. Define global availability as the preferred, alternate, or fallback load balancing mode.
5. List the virtual servers in descending order of preference. See Figure 6.6 for details.

A Global Availability configuration example

With the global availability load balancing mode, you can configure one data center as your primary service and have several alternate services on standby. In the **wideip** statement, list the virtual servers in descending order of preference. The first available virtual server is chosen for each resolution request.

Figure 6.6 shows a sample **wideip** definition where global availability is the preferred load balancing mode.

```
// Global availability
wideip {
   address           192.168.101.60
   port              80 // http
   name              "cgi.wip.domain.com"
   pool {
      name           "mypool"
      preferred      ga
      address       192.168.101.60
      address       192.168.102.60
      address       192.168.103.60
   }
}
```

Figure 6.6 *Configuring a standby data center*

The first listed virtual server (**192.168.101.60** in this example) receives all resolution requests unless it becomes unavailable. If the first listed virtual server does become unavailable, then the 3-DNS Controller sends resolution requests to the second listed virtual server, and so on.

Setting up load balancing for services that require multiple ports

Some sites require that you use multiple ports or services to access them. For these cases, you can configure a wide IP so that connections are not sent to a given address unless all specified ports or services are available.

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

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

To manually configure multiple ports for a wide IP

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Locate the **wideip** statement you want to edit.
4. Add the **port_list** line as indicated in bold in Figure 6.7.

```
wideip {
  address <ip_addr>
  port <port_number> | <"service name">
  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>
    } ]
  [ pool_lbmode <rr | ratio | ga | random | topology> ]
  [ pool definitions ...]
```

Figure 6.7 *Enabling the port_list option*

An example configuration for e-commerce services

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

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

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

Figure 6.8 *Syntax for e-commerce services*

For every virtual server address in the pool, a virtual server definition must exist for each port in the port list. For the example shown in Figure 6.8, the BIG-IP Controllers and host machines must have the following virtual servers defined:

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

Setting up topology-based features

The 3-DNS Controller supports two topology-based features: topology-based access control, and topology load balancing mode.

To use the topology load balancing mode, you should first set up topology-based access control so that the list records are defined. Otherwise, the topology load balancing mode has no effect.

Setting up topology-based access control is described next. Using the topology load balancing mode is described on page 6-23.

Setting up topology-based access control

You can use topology-based access control to implement a form of wide-area IP filtering. Topology-based access control allows you to specify which virtual servers are acceptable for a given resolution request, based on the proximity of the virtual server's IP address to the requesting LDNS server's IP address.

Understanding the list records

The ***list records*** in the topology definition define a score for pairs of known LDNS servers and virtual servers.

For example, here is a sample list record:

```
192.168.101.0/24   198.0.0.0/24     6
```

Essentially, each record defines two network endpoints in CIDR (Classless Inter-domain Routing) format, and a score. The CIDR format consists of an IP address and a number **n** designating a subnet bitmask. The bitmask is made up of **n** ones followed by **32 - n** zeros. For example, for **n = 8**, the bitmask is:

```
11111111000000000000000000000000
\______/\_____________________/
 8 ones          24 zeros
```

The first endpoint, **A**, corresponds to the IP address of a server (either a BIG-IP Controller or a host). The second endpoint, **B**, corresponds to the IP address of the LDNS. Suppose an LDNS, **L**, requests a name resolution from the 3-DNS Controller, and the virtual server being considered as an answer is managed by a BIG-IP Controller, **S**. The list record that matches is the one where the following equation is TRUE:

```
((S & A-mask == A & A-mask) && (L & B-mask == B & B-mask))
```

Referring to the example **topology** statement that starts on page 6-20, say that the LDNS **198.0.0.0** requested name resolution for **www.domain.com**, and a virtual server in the pool belonged to the BIG-IP Controller **192.168.101.0**. In this scenario, the 3-DNS Controller considers the first list record to be a match.

Note that in the above list record, the single ampersand (&) is a bitwise operator, and the double ampersands (&&) are logical operators.

Understanding the topology score

Each list record includes a score, which is used both in topology-based load balancing, and in topology-based access control. If multiple list records in a **topology** statement have the exact same server IP/mask and LDNS IP/mask but have different scores, only the last record is declared valid. For example, the first set of records is equivalent to the second set of records.

```
192.168.101.0/24   198.0.0.0/24     6
192.168.101.0/8    198.0.0.0/8      1
192.168.101.0/24   198.0.0.0/24     89 <-- replaces 1st record
192.168.101.0/24   198.0.0.0/24     0  <-- replaces previous record
192.168.101.0/24   198.0.0.0/24     3  <-- replaces previous record
```

This set of records is equivalent to the above set of five records.

```
192.168.101.0/8    198.0.0.0/8      1
192.168.101.0/24   198.0.0.0/24     3
```

Using the longest match rule

The 3-DNS Controller uses the same type of longest match rule that routers commonly use. If there are several IP/mask items that match a particular IP address, the 3-DNS Controller selects the record that is most specific, and thus has the longest mask (**n** is the largest).

For example, 192.168.101.4 matches 192.168.101.4/0, 192.168.101.4/8, 192.168.101.4/13, 192.168.101.4/24, and 192.168.101.4/32, but the longest matching IP/mask is 192.168.101.4/32. When the **longest_match** parameter is set to **yes** (the default), the longest match rule is obeyed for LDNS IP addresses, and also for server IP addresses, when there are multiple matches for a server/LDNS combination. This means that for the virtual server **192.168.101.50** owned by BIG-IP Controller **192.168.101.40** and LDNS **198.0.0.40**, the third list record is the longest match:

```
192.168.101.0/24   198.0.0.40/24    2
192.168.101.0/8    198.0.0.40/16    0
192.168.101.0/8    198.0.0.40/27    6 <-- Longest Match
192.168.101.0/16   198.0.0.0/24     7
192.168.101.0/32   198.0.0.0/24     3 <-- Second Longest Match
```

Although this is not how the search is implemented, consider that all the records matching the server and LDNS IP address are gathered into a set. The records in this set are sorted in descending order first by LDNS mask, and then by server mask. The highest record in the sorted set determines which is the shortest path between the client and a virtual server. For example, if the third list record in the above example is removed, then the first and fifth records tie for longest match on LDNS, but the fifth wins because it has the more specific server mask.

Implementing topology-based access control

Any virtual server/LDNS matching a list record with a score below the **acl_threshold** is interpreted as if the virtual server were unavailable. For example, if a LDNS **198.0.0.0** requests a name resolution, any virtual server in the class C subnet **192.168.101** is considered ***down*** for load balancing purposes due to the first list entry. This provides a hook for an administrator to set up access control to virtual servers based on LDNS IP addresses.

Explicitly allowing or denying access

You may want to define a wildcard list record that you can use to prevent users from being locked out when access control is turned on (when the **acl_threshold** is set to a value greater than zero). If the 3-DNS Controller compares the LDNS server's IP address to the specific list records but does not find a match, it can use a wildcard list record to determine how to handle the resolution request.

A wildcard list record is the last list record in the topology statement and uses the following syntax:

```
0.0.0.0/0      0.0.0.0/0 <score>
```

By using the subnet bitmask values **0** in the wildcard list record, this record will always be chosen last by the longest match rule.

The **<score>** parameter setting either allows or denies access, depending on whether its value is set greater than or less than the **acl_threshold** setting. A **<score>** value that is greater than or equal to the **acl_threshold** setting allows access. A **<score>** value that is less than the **acl_threshold** setting denies access.

If no wildcard list record is provided, the following is assumed:

```
0.0.0.0/0      0.0.0.0/0       0
```

Using access control to limit path probing

The **limit_probes** parameter specifies whether to apply access control to the probing of paths. If this parameter is set to **yes**, the 3-DNS Controller requests a particular BIG-IP Controller to probe only those LDNS servers that can connect to it according to the

probe_threshold value and the topology map scores. In the example **topology** statement that starts on page 6-20, the 3-DNS Controller would not send an LDNS **200.0.0.0** to the BIG-IP Controller **192.168.101.0** for probing, but would send it to the BIG-IP Controller **192.168.103.0**.

To configure topology-based probe access control using the Configuration utility

1. In the navigation pane, click **Topology**.
2. On the toolbar, click **Topology Settings**.
3. Add the topology settings. For help on configuring the topology settings, click **Help** on the toolbar.

To manually configure topology-based probe access control

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration** to open the **wideip.conf** file.
3. Place the **topology** statement at the end of the **wideip.conf** file.
4. Use the syntax shown in Figure 6.9 to define the **topology** statement. Also see the example that starts on page 6-20.

```
topology {
    acl_threshold   <0..4294967295>
    probe_threshold   <0..4294967295>
    limit_probes    <yes | no>
    longest_match   <yes | no>
    <server cidr> <LDNS cidr> <score>
}
```

Figure 6.9 *Syntax for topology statement*

An example configuration for topology access control

Suppose that your company maintains Spanish web sites. You have data centers in New York, Los Angeles, and Tokyo. You prefer that resolution requests made from clients located in North America are resolved by North American data centers. However, you do not mind if a few requests are sent to Tokyo when requests cannot be resolved in New York or Los Angeles.

Because of cost issues, you do not want requests from clients in South America to be sent to the New York data center. To achieve this, you can configure the **topology** statement as shown.

```
topology {
   acl_threshold 5
   probe_threshold 5
   limit_probes  yes
   longest_match yes

// server/mask   ldns/mask   score


//////////////////////////
// North American LDNS's:
//   198.0.0.0/8
//   199.0.0.0/8

// North America Priority List
//
// 1. New York
// 2. L.A.
// 3. Tokyo

   // New York
   192.168.101.0/24   198.0.0.0/8   30
   192.168.101.0/24   199.0.0.0/8   30

   // Los Angeles
   192.168.102.0/24   198.0.0.0/8   20
   192.168.102.0/24   199.0.0.0/8   20

   // Tokyo
   192.168.103.0/24   198.0.0.0/8   10
   192.168.103.0/24   199.0.0.0/8   10
```

Figure 6.10 *Example syntax for the topology statement (continued on next page)*

```
//////////////////////////
// South American LDNS's:
//   200.0.0.0/8
//   201.0.0.0/8

// South America Priority List
//
// 1. Tokyo
// 2. L.A.
// (New York excluded by acl_threshold)

   // Tokyo
   192.168.103.0/24   200.0.0.0/8   30
   192.168.103.0/24   201.0.0.0/8   30

   // Los Angeles
   192.168.102.0/24   200.0.0.0/8   20
   192.168.102.0/24   201.0.0.0/8   20

   // New York
   192.168.101.0/24   200.0.0.0/8   0
   192.168.101.0/24   201.0.0.0/8   0

//////////////////////////
// Wildcard List Record
//
// By default, if a list record is not found in the
// topology map for an LDNS, the score is assumed to
// be 0. By including the following "wildcard" list
// record, all other LDNS's (not North or South America
// as specified above) are assigned a score of 1 so
// the acl_threshold does not indicate that the
// virtual servers are down.

   0.0.0.0/0   0.0.0.0/0   1

}
```

Figure 6.10 *Example syntax for the topology statement (continued from previous page)*

Using the topology load balancing mode

The topology load balancing mode distributes connections based on the proximity of LDNS servers to particular data center. Proximity is determined by network IP addresses of the LDNS servers compared to network IP addresses of the virtual servers, and not necessarily by geographical location.

Configure topology access control before attempting to use the Topology load balancing mode. If the topology list records are empty, the virtual servers are load balanced as Global Availability.

To implement the topology load balancing mode using the Configuration utility

1. In the navigation pane, click **Wide IPs**.
2. In the Wide IP column, click a wide IP name.
   The Modify Wide IP screen opens.
3. On the toolbar, click **Modify Pool**.
   The Modify Wide IP Pools screen opens.
4. In the Pool Name column, click the name of a pool.
   The Modify Load Balancing screen opens.
5. Select **Topology** as the **Preferred**, **Alternate**, or **Fallback** load balancing mode.
6. Click **Update**.

To manually implement the topology load balancing mode

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

Figure 6.11 shows a sample **wideip** definition where topology is the preferred load balancing mode.

```
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 6.11 *Specifying load balancing according to topology score*

F5 Networks
3·DNS®
Distributed Traffic Controller

7

Monitoring and Administration

- Monitoring and administration utilities provided on the 3-DNS Controller
- Working with the 3-DNS Maintenance menu
- Changing passwords for the 3-DNS Controller
- Viewing system statistics

Monitoring and administration utilities provided on the 3-DNS Controller

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

The 3-DNS Controller provides the following configuration, monitoring, and administration utilities:

- **Configuration utility**
  The Configuration utility is a browser-based application you can use to configure and monitor the 3-DNS Controller. You may have used the Configuration utility to define your network setup. The Configuration utility supports Netscape Navigator, version 4.5 or later, and Internet Explorer, version 4.02 or later.
- **3-DNS Maintenance menu**
  The 3-DNS Maintenance menu is a command line utility you can use to manually configure the 3-DNS Controller. Use the 3-DNS Maintenance menu to simplify certain tasks such as starting the **big3d** agent and editing the **wideip.conf** file.

Working with the 3-DNS Maintenance menu

You can use the 3-DNS Maintenance menu to manually configure and monitor the 3-DNS Controller. However, if you work with either the browser-based Configuration utility or the NameSurfer application, you cannot use the 3-DNS Maintenance menu.

You can use the 3-DNS Maintenance menu to perform the following types of manual configuration tasks:

- Configure wide IPs
- View statistics
- Work with the **big3d** agent
- Manage synchronized files
- Work with security issues

- Configure the 3-DNS web server
- Work with **syncd**
- Configure NTP
- Configure NameSurfer

Figure 7.1 shows the 3-DNS Maintenance menu.

```
3 D N S(®)  Maintenance Menu

Generate RSA Authentication
Generate and Copy iQuery Encryption Key
Check versions of named, BIG-IP kernel and needed big3d
Edit big3d matrix
Install and Start big3d
Edit BIND Configuration
Edit 3-DNS Configuration
Synchronize Metrics Data
Check big3d
Restart big3d
Reconfigure 3-DNS Web Administration
Restart 3-DNS Administration
Change/Add Users for 3-DNS Web Administration
Dump and List named Database
Stop syncd
Restart syncd
Checkpoint synced files
Rollback checkpoint
Configure connection to NTP time server
Configure NameSurfer(TM)
Enter 'q' to Quit
```

Figure 7.1 *3-DNS Maintenance menu*

To use the 3-DNS Maintenance menu

1. Type the following command to open the menu:
   `3dnsmaint`
2. Select the command to execute, and press the Enter key.

Each command is described in the following sections.

Configuring wide IPs

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

Edit BIND Configuration

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

◆ Note

Use this command only if you are performing all configuration tasks manually. It is important that you do not use this command if you are using the Configuration utility or NameSurfer.

Edit 3-DNS Configuration

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

- Opens the **wideip.conf** file for editing
- Copies the **wideip.conf** file to all other 3-DNS Controllers in the local 3-DNS Controller's sync group
- Restarts **named**

Viewing statistics

Use the following command to view various 3-DNS Controller statistics:

Dump and List named Database

The **Dump and List named Database** command corresponds to the **3dprint** script, which 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, the name of the sync group to which each 3-DNS Controller belongs, and so on.
- **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, the number of times the 3-DNS Controller resolves requests to those virtual servers, and more.
- **Hosts**
  Displays statistics about all hosts known to the 3-DNS Controller such as the number of times 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 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, and so on.
- **Local DNS**
  Displays statistics collected for LDNS servers: the number of resolution requests received from a given server, the current protocol used to probe the server, and more.

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

To view more statistics information, click the Expand button (**+**) next to **Statistics** on the navigation pane in the Configuration utility.

Working with the big3d agent

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

Check versions of named, BIG-IP kernel and needed big3d

The **Check versions of named, BIG-IP kernel and needed big3d** command runs the **big3d_version** script. This script displays version numbers for all BIG-IP Controllers known to the 3-DNS Controller, and the version numbers of the **big3d** agent and **named** utility running on each BIG-IP Controller.

Edit big3d matrix

The **Edit big3d matrix** command opens an editable file that lists version numbers for all BIG-IP Controllers known to the 3-DNS Controller, and the version numbers of the **big3d** agent and **named** utility running on each BIG-IP Controller.

You do not need to edit this file unless a new BIG-IP kernel or a **named** version creates a conflict. If this happens, you need to place a new version of the **big3d** agent on all BIG-IP Controllers.

The **Install and Start big3d** command uses the matrix file to determine which version of the **big3d** agent to transfer.

Install and Start big3d

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

Check big3d

The **Check big3d** command runs the **big3d_check** script, which verifies that each BIG-IP Controller is running the **big3d** agent.

Restart big3d

The **Restart big3d** command runs the **big3d_restart** script, which stops and restarts the **big3d** agent on each BIG-IP Controller.

Managing synchronized files

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

Synchronize Metrics Data

The **Synchronize Metrics Data** command runs the **3dns_sync_metrics** script, which prompts you to copy metrics data from a remote 3-DNS Controller to the local 3-DNS Controller.

You should use this command only when you are configuring a new 3-DNS Controller.

Checkpoint synced files

The **Checkpoint synced files** command runs the **syncd_checkpoint** script, which creates a ***checkpoint file***. A checkpoint file is a compressed tar file that contains an archive of the files that are synchronized.

For more information, see *syncd_checkpoint*, on page B-9.

Rollback checkpoint

The **Rollback checkpoint** command runs the **syncd_rollback** script, which unrolls a checkpoint file. The checkpoint file contains the last saved copy of all files synchronized by **syncd**.

For more information, see *syncd_rollback*, on page B-10.

Working with security issues

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

Generate RSA Authentication

The **Generate RSA Authentication** command runs the **3dns_auth** script, which configures ssh access to any new 3-DNS Controller or BIG-IP Controller that is added to a network.

The **3dns_auth** script generates a password authentication by setting the **RSA Authentication** parameter to **yes** in **/etc/sshd_config.conf** and copying the **ssh** key to each 3-DNS Controller and BIG-IP Controller. When prompted for an RSA passphrase, press the Enter key instead of typing a password.

For more information, see *3dns_auth*, on page B-3.

Generate and Copy Encryption iQuery Key

The **Generate and Copy Encryption iQuery key** command runs the **install_key** script, which then runs the **F5makekey** script. **F5makekey** generates a seed key for encrypting communications between the 3-DNS Controller and BIG-IP Controller.

For more information, see *install_key and F5makekey*, on page B-9.

◆ Note

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

Using the 3-DNS web server

You can use the following commands to configure the 3-DNS web server.

Reconfigure 3-DNS Web Administration

The **Reconfigure 3-DNS Web Administration** command runs the **3dns_web_config** script, which lets you make configuration changes to the 3-DNS web server.

Restart 3-DNS Administration

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

Change/Add Users for 3-DNS Web Administration

The **Change/Add Users for 3-DNS Web Administration** command runs the **3dns_web_passwd** script, which 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.*

Working with syncd

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

Stop syncd

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

Restart syncd

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

Configuring NTP

The 3-DNS Controllers in a network must have their time synchronized to within a few seconds of each other. If you do not synchronize the controller, it is done by default through iQuery

messages exchanged between 3-DNS Controllers. However, the following command allows much more precise time synchronization between the 3-DNS Controllers.

Configure Connection to NTP Time Server

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

Configuring NameSurfer

You can use the following command to have NameSurfer handle DNS zone file management on the 3-DNS Controller.

Configure NameSurfer

The **Configure NameSurfer** command makes NameSurfer the master on the 3-DNS Controller, and NameSurfer then handles the zone file management, dealing with all changes and updates to the zone files. You can access the NameSurfer application in the Configuration utility by clicking **NameSurfer** on the navigation pane. (Note that if you do not set NameSurfer to be the master for your wide IP zones, you cannot use the Configuration utility. Instead, you must manually configure all 3-DNS Controller settings.)

Changing passwords for the 3-DNS Controller

The First-Time Boot utility prompts you to define a password that allows remote access to the 3-DNS Controller, and also prompts you to define a password for the 3-DNS Web server. You can change these passwords at any time.

To change the root user password for command line access

1. At the 3-DNS Controller command line prompt, log in as **root** and use the **passwd** command.
2. At the **password** prompt, type the password you want to use for the 3-DNS Controller and press Enter.
3. To confirm the password, retype it and press Enter.

Changing passwords and adding new user IDs for the 3-DNS web server

You can create new users for the 3-DNS web server, change a password for an existing user, or recreate the password file altogether, without actually going through the 3-DNS web server configuration process.

To add a new user ID using the Configuration utility

1. In the navigation pane, click **User Admin**.
   The User Administration screen opens.
2. Add the user administration settings. For help on configuring the settings, click **Help** on the toolbar.

To change or add user information using the 3-DNS Maintenance menu

1. At the command prompt, type **3dnsmaint** to open the 3-DNS Maintenance menu.
2. On the 3-DNS Maintenance menu, select the **Change/Add Users for 3-DNS Web Administration** command.

To create new users and change passwords for existing users manually

The following command creates a new user ID, or changes the password for an existing user ID. In place of the **<username>** parameter, type the user ID for which you want to create a password:

```
/var/f5/httpd/bin/htpasswd /var/f5/httpd/basicauth/users \
   <username>
```

Once you enter the command, you are prompted to type the new password for the named user.

To manually create a new password file

The following command recreates the 3-DNS web server password file, and defines one new user ID and password. In place of the **<username>** parameter, type the user ID that you want to create:

```
/var/f5/httpd/bin/htpasswd -c /var/f5/httpd/basicauth/users \
   <username>
```

Once you enter the command, you are prompted to type the new password for the named user.

Viewing system statistics

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

- Global settings
- Disabled objects
- Virtual connections between LDNS servers and virtual servers for given wide IPs
- Data centers
- Sync groups
- Wide IPs

- 3-DNS Controllers
- BIG-IP Controllers
- Probers
- Other host machines
- Virtual servers
- Paths
- Local DNS servers

To view system statistics

1. In the navigation pane, click the Expand button (+) next to **Statistics**.
2. From the list, select the item representing the statistics you wish to view.
3. For details on what kind of information the various statistics pages are displaying, click **Help** on the toolbar of the specific statistics page you are viewing.

F5 Networks

3·DNS®

Distributed Traffic Controller

8

Configuring SNMP

- Working with SNMP on the 3-DNS Controller

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 F5 Networks 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 manage the 3-DNS Controller by configuring traps for the SNMP agent or by polling the controller with your standard network management station (NMS).

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.

To securely manage information that is collected by the 3-DNS SNMP agent, you have the following security options available:

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

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/rc.local**

- **/etc/snmptrap.conf**
- **/etc/syslog.conf**
- **checktrap.pl**

❖ 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 may 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 F5 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 provides standard management information.

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****, later in this chapter.*

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/host.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 subnetwork 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

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****, later in this chapter.*

The **/etc/snmpd.conf** file controls most aspects of the SNMP agent. This file is used to set up and configure certain traps, passwords, and general SNMP variable names. A few of the necessary variables are listed below:

- **System Contact Name**
  The System Contact is a MIB-II simple string variable defined by almost all SNMP 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 almost all boxes support. 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)**.
- **authtrapenable <integer>**
  Set this variable to **1** so that traps can be sent for authentication warnings. Set the variable to **2** to disable it.

◆ **Note**

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

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

/etc/rc.local

The following entry in the **/etc/rc.local** file sets the SNMP agent to automatically start up when you boot the 3-DNS Controller (Figure 8.1).

```
# 3DNS SNMP Agent
if [ -f /etc/snmpd.conf ]; then
   /sbin/snmpd -c /etc/snmpd.conf
fi
```

Figure 8.1 *Starting the SNMP agent in the* ***/etc/rc.local*** *file.*

If the **/etc/snmpd.conf** file is present on your system, the SNMP agent starts automatically.

/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 8.2

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

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

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

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

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

Figure 8.2 *Excerpt from the* ***/etc/snmptrap.conf*** *file*

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

- Virtual server green to red
- Virtual server red to green
- Server green to red
- Server red to green
- CRC failure

To see messages that are triggering a 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 against 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.

F5 Networks 3·DNS® Distributed Traffic Controller

9

Controlling Network Traffic Patterns

- 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 9-7, 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 9-6.

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 includes:

- **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 and also available in Appendix A).

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 the following guidelines:

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

- **3dscript** 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 parenthesis
  - A unary-operator followed by a conditional-expression enclosed in parenthesis

- 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 9-5 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. Other production rule examples are included in the sample **wideip.conf** file installed on the 3-DNS Controller (and available in Appendix A). 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 9.1 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 9.1 *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 9.2 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 9.2 *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 9.3 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 9.3 *Sending a hacker to a specific server*

A related example, shown in Figure 9.4, 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 9.4 *Detecting an iQuery failure due to potential attack*

F5 Networks

3·DNS®

Distributed Traffic Controller

A

Wideip.conf Syntax

Overview of the wideip.conf file

The 3-DNS Controller configuration file is called **/etc/wideip.conf**. The **wideip.conf** file describes a network's data centers, servers (3-DNS Controllers, BIG-IP Controllers, and hosts), and wide IPs. The **wideip.conf** file consists of two types of information: statements and comments.

Your **wideip.conf** file should include at least the following definitions.

- A **datacenter** statement. If you do not create one, the 3-DNS Controller creates one for each configured server and names each as follows: **generic-<server IP address>**
- At least one virtual server, which is defined as part of a BIG-IP or host **server** statement
- At least one **server** statement defining a 3-DNS Controller
- A **wideip** statement

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

- If the file cannot be parsed, **named** will not start.
- If the file can be parsed, 3-DNS Controllers revert to standard DNS behavior.

To open the /etc/wideip.conf file

On the 3-DNS Maintenance menu, select **Edit 3-DNS Configuration**.

WARNING

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

Using include files

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

Using include files reduces the size of the **wideip.conf** file and makes it easier to manage your configuration. In addition, include files lead to better system performance because metrics for each aspect of your network are collected and dumped only to the relevant file, instead of having all metrics collected and dumped in a single, potentially unwieldy, file.

Include files are automatically created and implemented whenever one of the following occurs:

- You configure your network setup using the Configuration utility.
- You perform any type of dumping operation. By default dumping operations are **on**. To turn dumping **off**, set the **global** sub-statement **timer_persist_cache** to **0**.

◆ Note

When include files are generated, any comments you incorporated are deleted.

To see an example of a **wideip.conf** file that incorporates include files, see *Sample 3-DNS Controller configuration file using include files*, on page A-65.

Syntax for include files

The following syntax is used when incorporating include files into a **wideip.conf** file.

```
include root_in "/var/3dns/include"
include root_out "/var/3dns/include"
include global <"file_name.conf">
include datacenter <"file_name.conf">
include sync_group <"file_name.conf">
include server <"file_name.conf">
include wideip <"file_name.conf">
include 3dscript <"file_name.conf">
include topology <"file_name.conf">
```

Figure A.1 *Special syntax for include files*

Definition of include statements

Parameter	Description	Default
include root_in	Specifies the name of the default directory. Enclose all file names in quotation marks.	**"/var/3dns/include"**
include root_out	Specifies the name of the default directory.	**"/var/3dns/include"**
include global	Specifies the name of the file that contains the **globals** statement.	**"globals.conf"**
include datacenter	Specifies the name of the file that contains **datacenter** statements.	**"datacenters.conf"**
include sync_group	Specifies the name of the file that contains **sync_group** statements.	**"sync_groups.conf"**
include server	Specifies the name of the file that contains **server** statements.	**"servers.conf"**
include wideip	Specifies the name of the file that contains **wideip** statements.	**"wideips.conf"**
include 3dscript	Specifies the name of the file that contains production rule configuration.	**"prodrules.conf"**
include topology	Specifies the name of the file that contains the **topology** statement.	**"topology.conf"**
include manifest	Specifies the name of the file that the Configuration utility uses to manage any production rules generated by the utility. It is important that you **do not** edit this statement.	**"/var/f5/www/seeit/.prodruledb/manifest"**

Table A.1 *Include file descriptions*

Statements

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

You can find an example of a complete configuration file in *Sample 3-DNS Controller configuration file*, on page A-48.

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

- **globals**
  Controls global 3-DNS Controller configuration options and sets defaults for other statements.
- **datacenter**
  Defines the group of 3-DNS Controllers, BIG-IP Controllers, and hosts that reside in a single physical location.
- **sync_group**
  Defines the group of 3-DNS Controllers that synchronize their configuration settings and metrics data.
- **server**
  Defines a 3-DNS Controller, a BIG-IP Controller, or a generic host machine.
- **wideip**
  Defines a wide IP. Wide IPs map a domain name to a load balancing mode and a set of virtual servers (on BIG-IP Controllers and other host machines).
- **topology**
  Implements and defines topology-based access control, and makes it possible for you to use the topology load balancing mode (on its own and as part of the QOS mode).

Syntax rules

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

- **Statement order**
  Statements should appear in this order:
  - **globals**
  - **datacenter**
  - **sync_group**
  - **server**
  - **wideip**
  - **topology**

- **Port specification**
  When you define 3-DNS Controllers or hosts, the port specification must follow the **probe_protocol** sub-statement. When you define a BIG-IP Controller or virtual server (on a BIG-IP Controller or on a host), the port specification must immediately follow the address specification and can take any of the following forms:

```
address <ip_addr>:<port>
address <ip_addr>
    port <port>
address <ip_addr>
    service <wks>
```

In the above example, **<wks>** stands for well-known service and is a quoted string representing the name of a WKS defined in the **/etc/services** file.

- **cur_ values**
  You may notice several **cur_** values in your **wideip.conf** file; do not edit them unless you are instructed to do so by your vendor's technical support. For more information, see *Understanding cur_ values*, on page A-79.

Typography in syntax examples

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

- **Mandatory parameters**
  Angle brackets (< >) enclose mandatory parameters where you must type the data associated with a command.
- **Optional parameters**
  Brackets ([ ]) enclose optional parameters.
- **Choice of parameters**
  A vertical bar (|) between two values means that either value is acceptable.

The globals statement

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

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

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

Figure A.2 *Multiple globals sub-statements*

Syntax for the globals statement

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

```
globals {
  [ time_tolerance <number> ]
  [ encryption < yes | no > ]
  [ encryption_key_file <string> ]
  [ check_static_depends < yes | no > ]
  [ check_dynamic_depends < yes | no > ]
  [ persist_ldns < yes | no > ]
  [ timer_get_3dns_data <number> ]
  [ timer_get_bigip_data <number> ]
  [ timer_get_host_data <number> ]
  [ timer_get_vs_data <number> ]
  [ timer_get_path_data <number> ]
  [ timer_get_trace_data <number> ]
  [ timer_check_keep_alive <number> ]
  [ timer_persist_cache <number> ]
  [ 3dns_ttl <number> ]
  [ bigip_ttl <number> ]
  [ host_ttl <number> ]
  [ vs_ttl <number> ]
  [ path_ttl <number> ]
  [ trace_ttl <number> ]
  [ default_ttl <number> ]
  [ rtt_timeout <number> ]
  [ rtt_sample_count <number> ]
  [ rtt_packet_length <number> ]
  [ rx_buf_size <number> ]
  [ tx_buf_size <number> ]
  [ qos_coeff_rtt <number> ]
  [ qos_coeff_completion_rate <number> ]
  [ qos_coeff_packet_rate <number> ]
```

Figure A.3 *Syntax for the globals statement (continued on next page)*

```
[ qos_coeff_topology <number> ]
[ qos_coeff_hops <number> ]
[ qos_coeff_vs_capacity ]
[ qos_factor_rtt <number> ]
[ qos_factor_completion_rate <number> ]
[ qos_factor_packet_rate <number> ]
[ qos_factor_topology <number> ]
[ qos_factor_hops <number> ]
[ qos_factor_vs_capacity <number> ]
[ default_alternate < ga | null | random | ratio |
  return_to_dns | rr | topology | vs_capacity > ]
[ default_fallback < completion_rate | ga | hops | leastconn |
  null | packet_rate | qos | random | ratio | return_to_dns |
  rr | rtt | topology | vs_capacity> ]
[ fb_respect_depends < yes | no > ]
[ fb_respect_acl < yes | no > ]
[ path_duration <number> ]
[ ldns_duration <number> ]
[ prober <ip_addr> ]
[ resolver_tx_buf_size <number> ]
[ resolver_rx_buf_size <number> ]
[ use_alternate_iq_port < yes | no > ]
[ multiplex_iq < yes | no > ]
[ paths_never_die < yes | no > ]
[ paths_noclobber < yes | no > ]
[ rtt_probe_dynamic < yes | no > ]
[ rtt_port_discovery < yes | no > ]
[ rtt_autorecover_discovery < yes | no > ]
[ rtt_discovery_method < short | wks | full | all > ]
[ rtt_allow_probe < yes | no > ]
[ rtt_allow_hops < yes | no > ]
[ rtt_allow_frag < yes | no > ]
```

Figure A.3 *Syntax for the globals statement (continued from previous page)*

```
  [ probe_protocol {
      [dns_ver]
      [dns_dot]
      [icmp]
      [udp]
      [tcp]
    }]
  [ datasize_system ]
  [ datasize_reap_pct ]
  [ default_iquery_protocol ]

}
```

Figure A.3 *Syntax for the globals statement (continued from previous page)*

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

```
globals {
  prober 192.168.101.2       // Default prober is New York 3DNS
  encryption yes             // Encrypt iQuery
  paths_noclobber yes        // Don't overwrite metrics with
                             // zeroed results
  path_ttl 2400              // Extend the life of path metrics
  rtt_probe_dynamic yes      // Switch probing method if current
                                fails
}
```

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

Definition of globals sub-statements

Each **globals** sub-statement supports the parameters described below.

Synchronization

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

Parameter	Description	Default
time_tolerance	Specifies the variation of time allowed (in seconds) when comparing time stamps on files. The **syncd** daemon allows for slight variation in time stamps when it compares files during the synchronization process. If the difference between the two time stamps falls within the **time_tolerance** setting, the daemon considers the files to be the same and does not overwrite one with the other. See *Setting the time tolerance value*, on page 4-32.	**10**

Table A.2 *Synchronization sub-statements*

Encryption

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

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

Table A.3 *Encryption sub-statements*

Dependencies

The dependencies sub-statement specifies whether the 3-DNS Controller checks the availability of virtual servers on BIG-IP Controllers or hosts before the 3-DNS Controller sends a connection to the virtual server.

Parameter	Description	Default
check_static_depends	Specifies whether to check the availability of virtual servers on BIG-IP Controllers and hosts. Change this option to **no** if you want to test your configuration. Settings this option to no has the effect of forcing the virtual servers to have green (up) status indicators on the virtual servers statistics page in the Configuration utility.	**yes**

Table A.4 *Dependencies sub-statement*

LDNS Persistence

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

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

Table A.5 *LDNS persistence*

Periodic task intervals

These sub-statements define the frequency at which the 3-DNS Controller refreshes the metrics information it collects.

Parameter	Description	Default
timer_get_3dns_data	Specifies how often the 3-DNS Controller retrieves performance data for other 3-DNS Controllers in the sync group. You can enter a value between **0** and **4294967295** seconds.	**20**
timer_get_bigip_data	Specifies how often the 3-DNS Controller refreshes BIG-IP Controller information. You can enter a value between **0** and **4294967295** seconds.	**20**
timer_get_host_data	Specifies how often the 3-DNS Controller refreshes other host machine information. You can enter a value between **0** and **4294967295** seconds.	**90**
timer_get_vs_data	Specifies how often the 3-DNS Controller refreshes virtual server information. You can enter a value between **0** and **4294967295** seconds.	**30**
timer_get_path_data	Specifies how often the 3-DNS Controller refreshes path information (for example, round trip time or ping packet completion rate). You can enter a value between **0** and **4294967295** seconds.	**120**
timer_get_trace_data	Specifies how often the 3-DNS Controller retrieves traceroute data (traceroutes between each data center and each LDNS). You can enter a value between **0** and **4294967295** seconds.	**60**
timer_check_keep_alive	Specifies how often the 3-DNS Controller queries remote 3-DNS Controllers and BIG-IP Controllers. This value determines how often **named** sends hello packets to each **big3d** agent in its configuration. You can enter a value between **0** and **4294967295** seconds.	**60**
timer_persist_cache	Specifies how often the 3-DNS Controller writes the **wideip.conf** file from memory. You can enter a value between **0** and **4294967295** seconds.	**300**

Table A.6 *Periodic task interval statements*

Data timeouts

These sub-statements set the amount of time for which metrics information is considered valid. After a timeout is reached, the 3-DNS Controller refreshes the information.

Parameter	Description	Default
default_ttl	Specifies the default number of seconds that the 3-DNS Controller considers the wide IP **A** record to be valid. If you do not specify a wide IP TTL value when defining a wide IP, the wide IP definition uses the **default_ttl** value.	**30**
3dns_ttl	Specifies the number of seconds that the 3-DNS Controller considers performance data for the other 3-DNS Controllers to be valid.	**60**
bigip_ttl	Specifies the number of seconds that BIG-IP Controller information is to be used by the 3-DNS Controller for name resolution and load balancing. You can enter a value between **1** and **4294967295**. The following relationship should be maintained: **bigip_ttl > timer_get_bigip_data**. A 2:1 ratio is the optimal setting for this relationship.	**60**
host_ttl	Specifies the number of seconds that generic host machine information is to be used by the 3-DNS Controller for name resolution and load balancing. You can enter a value between **1** and **4294967295**. The following relationship should be maintained: **host_ttl > timer_get_host_data**.	**240**
vs_ttl	Specifies the number of seconds that virtual server information (data acquired from a BIG-IP Controller or other host machine about a virtual server) is to be used by the 3-DNS Controller for name resolution and load balancing. You can enter a value between **1** and **4294967295**. The following relationship should be maintained: **vs_ttl > timer_get_vs_data**.	**120**
path_ttl	Specifies the number of seconds that path information is to be used by the 3-DNS Controller for name resolution and load balancing. You can enter a value between **1** and **4294967295**. The following relationship should be maintained: **path_ttl > timer_get_vs_data**.	**2400**
trace_ttl	Specifies the amount of time (in seconds) that the 3-DNS Controller considers traceroute data to be valid. You can enter a value between **1** and **4294967295**.	**604800** (seven days)

Table A.7 *Data timeouts sub-statements*

Metrics collection

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

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

Table A.8 *Metrics collection sub-statements*

Resource limits

The resource limits sub-statements define the amount of memory allocated to sending and receiving metrics information.

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

Table A.9 *Resource limits sub-statements*

QOS values

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

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

Figure A.5 *QOS equation*

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

Parameter	Description	Default
qos_coeff_rtt	Specifies the relative weighting for round trip time when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	**20**
qos_coeff_completion_rate	Specifies the relative weighting for ping packet completion rate when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	**5**
qos_coeff_packet_rate	Specifies the relative weighting for BIG-IP Controller packet rate when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	**3**

Table A.10 *QOS values sub-statements*

Parameter	Description	Default
qos_coeff_topology	Specifies the relative weighting for topology when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	**0**
qos_coeff_hops	Specifies the relative weighting for hops when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	**0**
qos_coeff_vs_capacity	Specifies the relative weighting for vs capacity when the load balancing mode is set to Quality of Service. You can enter a value between **0** and **4294967295**.	**0**
qos_factor_rtt	Specifies the factor used to normalize raw RTT values when computing the QOS score.	**10000**
qos_factor_completion_rate	Specifies the factor used to normalize raw completion rate values when computing the QOS score.	**10000**
qos_factor_packet_rate	Specifies the factor used to normalize raw packet rate values when computing the QOS score.	**10000**
qos_factor_topology	Specifies the factor used to normalize raw topology values when computing the QOS score.	**10**
qos_factor_hops	Specifies the factor used to normalize raw hops values when computing the QOS score.	**25**
qos_factor_vs_capacity	Specifies the factor used to normalize raw vs capacity values when computing the QOS score.	**1**

Table A.10 *QOS values sub-statements*

Load balancing

Parameter	Description	Default
default_alternate	Defines the default load balancing mode used for the alternate method (formerly **default_static**). You can override this setting in the **wideip** statement.	**rr**
default_fallback	Defines the default load balancing mode used for the fall-back method. You can override this setting in the **wideip** statement.	**return_to_dns**
fb_respect_depends	Determines whether the 3-DNS Controller respects virtual server status when load balancing switches to the specified fallback mode.	**no**
fb_respect_acl	Determines whether the 3-DNS Controller imposes topology access control when load balancing switches to the specified fallback mode.	**no**

Table A.11 *Load balancing sub-statements*

Prober

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

Parameter	Description	Default
prober	Specifies the default prober for host status, usually the 3-DNS Controller IP address. Using this sub-statement is not necessary if the 3-DNS Controller only manages the BIG-IP Controller. When this option is set to **0**, the 3-DNS Controller's IP address is the implied value. This sub-statement can be overridden within the **server** statement.	**0.0.0.0**

Table 1.12 *Prober sub-statement*

WARNING

You must define a prober if the 3-DNS Controller manages virtual servers on hosts. If you do not define a default prober in the globals, or probers in the host statements for all hosts, you will encounter validation errors.

Buffer size

The buffer size sub-statements specify the maximum amount of UDP data that the 3-DNS Controller can receive, and also specify the maximum amount of TCP data that the 3-DNS Controller can send.

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

Table A.13 *Buffer size sub-statements*

Reaping

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

The default reaping values are adequate for most configurations. Contact F5 technical support if you want to make changes to them.

Parameter	Description	Default
datasize_system	Specifies the amount of RAM the 3-DNS Controller reserves for system usage, such as non-3-DNS specific processes.	if 64 MB RAM, default is **32;** otherwise default is **64**
datasize_reap_pct	Specifies what percentage of memory that the 3-DNS Controller frees up during the reap process.	**15**
path_duration	Specifies the number of seconds that a path remains in the cache after its last access. You can type a value between **60** and **2147483648**).	**604800** (7 days)
ldns_duration	Specifies the number of seconds that an inactive LDNS will remain in the cache. Each time an LDNS makes a request, the clock starts again. You can type a value between **60** and **2147483648**.	**2419200** (28 days)

Table A.14 *Reaping sub-statements*

iQuery port options

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

Table A.15 *iQuery port options sub-statements*

Probing

Parameter	Description	Default
paths_never_die	Specifies that dynamic load balancing modes can use path data even after the TTL for the path data has expired. We recommend that you change this setting to **yes**, which has the effect of requiring that the 3-DNS Controller always uses path data even if the path's TTL expires.	**no**
paths_noclobber	Specifies whether the 3-DNS Controller overwrites existing path data with blank data when a path probe fails. With the default setting, the 3-DNS Controller does not overwrite existing path data with blank data when a path probe fails. Unlike **paths_never_die**, this parameter has no effect on **path_ttl**.	**yes**
check_dynamic_depends	Specifies that the 3-DNS Controller checks the availability of a path before it uses the path for load balancing. Changing this option to **no** overrides the **path_ttl** and whether the last probe attempt was successful. You can use this parameter in conjunction with **paths_noclobber**. This parameter does not prevent the refreshing of path metrics.	**yes**
rtt_port_discovery	Determines whether the 3-DNS Controller uses the discovery factory to find an alternate port to be used by the probing factory, if probing on port **53** fails.	**no**
rtt_autorecover_discovery	Specifies whether to move collected LDNS information from the Needs Discovery probing state to the Needs Probe state if **rtt_port_discovery** is set to **no**. Setting this to **no** means that if probing failed for a specific LDNS for any reason, the LDNS is ineligible for future probing attempts.	**yes**
rtt_discovery_method	Determines which ports to scan. The default, **short**, causes the 3-DNS Controller to scan a pre-defined list of ports, and then scans port **53**. Other acceptable values are **wks** (well-known services), **full** (all ports between **1** and **1024**), and **all**.	**short**

Table A.16 *Probing sub-statements*

The datacenter statement

A **datacenter** statement defines the group of 3-DNS Controllers, BIG-IP Controllers, and hosts that reside in a single physical location.

Syntax for the datacenter statement

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

```
datacenter {
  name <"data center name">
  [ location <"location info"> ]
  [ contact <"contact info"> ]
  [ 3dns <3DNS IP address> ]
  [ bigip <BIG-IP IP address> ]
  [ host <host IP address> ]
}
```

Figure A.6 *Syntax for the datacenter statement*

Figure A.7 shows an example of a valid **datacenter** statement:

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

Figure A.7 *Example syntax for the datacenter statement*

Definition of datacenter sub-statements

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

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

Table A.17 *Data center sub-statements*

The sync_group statement

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

Syntax for the sync_group statement

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

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

Figure A.8 *Syntax for the sync_group statement*

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

Figure A.9 shows an example of a valid **sync_group** statement:

```
sync_group {
  name "sync"
  3dns 192.168.101.2    // New York
  3dns 192.168.102.2    // Los Angeles
}
```

Figure A.9 *Example syntax for the sync_group statement*

Definition of sync_group sub-statements

Parameter	Description
name	Specifies the name of this sync group.
3dns	Specifies the IP address or domain name (enclosed in quotation marks) of a 3-DNS Controller in the group. First list the IP address of the principal itself. Then list all other 3-DNS Controllers, in the order that they should become principals should previously listed 3-DNS Controllers fail.

Table A.18 *Sync_group sub-statements*

The server statement

The **server** statement defines the characteristics associated with a particular 3-DNS Controller, BIG-IP Controller, or host.

Note

*The **server** statement replaces the **bigip** and **host** statements used in earlier versions of 3-DNS Controller. Although this version of 3-DNS Controller provides backward compatibility with the earlier **bigip** and **host** statements, we recommend that you use the newer syntax.*

A **server** statement contains the following information:

- The type of server: 3-DNS Controller, BIG-IP Controller, or host
- The IP address of the server
- If the server is a BIG-IP Controller or host, the set of virtual servers that are available on it
- If the server is a BIG-IP Controller, dynamically collected information about the BIG-IP Controller, its virtual servers and ports, and the paths between the BIG-IP Controller and LDNS

Because available sub-statements vary by server type, the syntax and examples for each type are listed separately. All sub-statements are defined in the table starting on page A-32.

Syntax for the server statement (3-DNS Controller)

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

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

Figure A.10 *Syntax for defining a 3-DNS Controller*

Figure A.11 shows an example of the syntax to use in defining a 3-DNS Controller:

```
// New York
server {
   type 3dns
   address 192.168.101.2
   name "3dns-newyork"
   iquery_protocol  udp
   remote {
      secure no
      user "root"
   }
   probe_protocol icmp
   port 53
}
```

Figure A.11 *Example syntax for defining a 3-DNS Controller*

Syntax for the server statement (BIG-IP Controller)

The following **server** statement syntax applies to BIG-IP Controllers only.

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

  vs {
    address <virtual server IP address>
    port <port number> | service <"service name">
    [ translate {
        address <IP address>
        port <port number>|service <"service name">
      } ]
   }
}
```

Figure A.12 *Syntax for defining a BIG-IP Controller*

Figure A.13 shows an example of the syntax to use in defining a BIG-IP Controller:

```
server {
   type            bigip
   address         192.168.101.40
   name            "bigip-newyork"
   iquery_protocol  udp

   remote {
      secure       yes
      user         "administrator"
   }

   # Tell 3-DNS about the 2 interfaces on a BIG-IP HA
   interface {
      address      192.168.101.41
      address      192.168.101.42
   }

   # Change the number of factories doing the work at big3d
   factories {
      prober     6
      discovery  1
      snmp       1
      hops       2
   }

   vs {
      address         192.168.101.50
      service         "http"
      translate {
         address      10.0.0.50
         port         80
      }
   }
   vs {
      address         192.168.101.50:25 // smtp
      translate {
         address      10.0.0.50:25
      }
   }
}
```

Figure A.13 *Example syntax for defining a BIG-IP Controller*

Syntax for the server statement (host)

The following **server** statement syntax applies to hosts only. Note that the **snmp** sub-statement is only necessary if you want the **big3d** agent to use an SNMP agent on the host to collect additional metrics information. For more information, see *Configuring host SNMP settings*, on page 4-17.

```
server {
  type host
  address <IP address>
  name <"host_name">
  probe_protocol <tcp | icmp | udp | dns_ver| dns_dot>
  [ prober <IP address> ]
  port <port number> | service <"service name">
  [ snmp {
    agent <generic | ucd | solstice | ntserv | ciscold | 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 | udp| dns_ver| dns_dot> ]
  }
}
```

Figure A.14 *Syntax for defining a host*

Figure A.15 shows an example of the syntax to use in defining a host:

```
server {
   type            host
   address         192.168.104.40
   name            "host-tokyo"
   probe_protocol  icmp
   port            53
   snmp {
      agent        ucd
      community    "public"
      version      1
   }
   vs {
      address         192.168.104.50:25
   }
   vs {
      address         192.168.104.50:80
   }
}
```

Figure A.15 *Example syntax for defining a host*

Definition of server sub-statements

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

Address information

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

Parameter	Description
type	Indicates whether the specified server is a 3-DNS Controller, BIG-IP Controller, or host.
address	Specifies the IP address of the 3-DNS Controller, BIG-IP Controller, or host.
name	Specifies the name of the 3-DNS Controller, BIG-IP Controller, or host. You must enclose all names in quotation marks.
iQuery_protocol	Specifies the iQuery transport option, TCP or UDP.
probe_protocol	Specifies the protocol method to use for probing this host: ICMP, UDP, or TCP. Applies to 3-DNS Controllers and hosts. Note that UDP is not supported on hosts.
prober	Specifies the IP address of the machine probing the 3-DNS Controller or host. This IP address points to either a BIG-IP Controller or a 3-DNS Controller that runs the **big3d** agent. The **big3d** agent actually probes the host and virtual servers to verify whether the host or a particular virtual server is currently available to accept connections. If you omit this parameter, the 3-DNS Controller uses the **prober <ip_addr>** parameter defined in the **globals** statement. Applies to 3-DNS Controller and hosts.
port	Specifies the port used to probe this host if **probe_protocol** is set to TCP. Applies to 3-DNS Controllers and hosts.

Table A.19 *Address information sub-statements*

Remote connections

Using the **remote** sub-statement is only necessary if you want to specify a different log-on name or specifically use SSH or RSH on 3-DNS Controllers and BIG-IP Controllers.

Parameter	Description
remote	Indicates the start of a remote sub-statement. Applies to 3-DNS Controllers and BIG-IP Controllers.
secure	Specifies whether to use **ssh** (secure shell) or **rsh** (remote shell) for remote connections. The default for US controllers is **yes**, which specifies that **ssh** is used. International versions must use **rsh** instead. Applies to 3-DNS Controllers and BIG-IP Controllers.
user	Specifies the "superuser" name that is used to allow a remote user to log on to the controller. Enclose this name in quotation marks. If you omit this parameter, the default, **"root"**, is used. Applies to 3-DNS Controllers and BIG-IP Controllers.

Table A.20 *Remote connections sub-statements*

Hardware redundancy

If you have hardware-redundant 3-DNS Controllers and BIG-IP Controllers, you must configure the **interface** sub-statement for the 3-DNS Controller to work properly with BIG-IP Controllers in Active-Active mode. This sub-statement is also required in using the standby BIG-IP Controller or 3-DNS Controller for probing.

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

Table A.21 *Hardware redundancy sub-statements*

Factories

With 3-DNS Controllers and BIG-IP Controllers, you can change the number and types of probing factories by using the **factories** sub-statement. If you omit this sub-statement, the defaults are used. For more information on probing, see *Setting up data collection with the big3d agent*, on page 2-9.

Parameter	Description
factories	Indicates the start of the factories definition. Applies to 3-DNS Controllers and BIG-IP Controllers.
prober	Specifies the number of prober factories to use.
discovery	Specifies the number of discovery factories to use.
snmp	Specifies the number of snmp factories to use.
hops	Specifies the number of hops factories to use.

Table A.22 *Factories sub-statements*

SNMP settings

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

If you need help configuring the SNMP agent on the host itself, see *Configuring SNMP agents on hosts*, on page 4-20.

Parameter	Description
snmp	Specifies the start of an SNMP definition. Applies to hosts only.
agent	Specifies the SNMP agent type. If you omit this parameter, the **big3d** agent uses the generic SNMP agent. Applies to hosts only.
port	Specifies the port the SNMP agent runs on. Applies to hosts only.
community	Specifies the password for basic SNMP security and for grouping SNMP hosts. Enclose this string in quotation marks. Applies to hosts only.

Table A.23 *SNMP sub-statements*

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

Table A.23 *SNMP sub-statements*

Virtual server definitions

Part of defining a BIG-IP Controller server or host server is defining the virtual servers that the server manages. After you define a virtual server here (including specifying the address and port), you can use this virtual server in a **wideip** definition.

Parameter	Description
vs	Indicates the start of a virtual server definition. Applies to BIG-IP Controllers and hosts.

Table A.24 *Virtual server definitions*

Parameter	Description
address	Specifies the IP address of a virtual server owned by a BIG-IP Controller or host. Note that the virtual server's address must be listed first, before port or service values. Applies to BIG-IP Controllers and hosts.
port or service	Specifies the virtual server's port number or service name. You can add the port number, preceded by a colon, on the same line as the virtual server's address, or you can enter it on the next line. You can use the service name if it is a **WKS** (well known service) and you enclose it in quotation marks. Applies to BIG-IP Controllers and hosts.
translate	Specifies that iQuery packets sent to the BIG-IP Controller include translated IP addresses (required if the packets must pass through a firewall). When you use this keyword, you must then include address and port/service information for the translated IP addresses. Applies to BIG-IP Controllers only.

Table A.24 *Virtual server definitions*

The wide IP statement

The **wideip** statement defines a wide IP. A wide IP maps a domain name to a load balancing mode and a set of virtual servers (on BIG-IP Controllers and/or other host machines).

Syntax for the wideip statement

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

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

Figure A.16 *Syntax for the wideip statement*

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

```
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 A.17 *Example syntax for the wideip statement*

Definition of wideip sub-statements

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

Address information

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

Parameter	Description
address	Specifies a key that represents one valid virtual server IP address from the set that services this wide IP. This key is also listed as the **A** record in the zone file for the domain.
port or service	Specifies the virtual server's default port number or service name. You can use the service name if it is a **WKS** (well known service) and you enclose it in quotation marks.
name	Specifies the domain name for this wide IP (for example, **"www.wip.domain.com"**). You must enclose all names in quotation marks.
alias	Specifies an alternate name for this wide IP. You must enclose all names in quotation marks. Alias names are treated the same as the domain name. You can specify up to 200 alias names for each wide IP.
ttl	Specifies the amount of time (in seconds) that the **A** record is used by the LDNS after resolving the wide IP. This is the TTL associated with the **A** record as specified by RFC 1035, on pages 29 and 30.
port_list	Specifies a list of ports that must be available before the 3-DNS Controller can send connections to the specified address.
persist	Specifies whether to maintain a persistent connection between an LDNS and a particular virtual server in this wide IP (rather than load-balancing the connection to any available virtual server).
persist_ttl	Specifies the number of seconds to maintain a persistent connection between an LDNS and a particular virtual server in this wide IP; this setting is valid only if you have configured the **persist** parameter.
qos_coeff	Specifies the relative weighting for each load balancing method in calculating the Quality of Service (QOS) mode. Before adjusting QOS coefficients, see *Understanding QOS coefficients*, on page 6-2.
pool_lbmode	Specifies the load balancing mode to use to balance requests over all pools.
pool	Indicates the start of the pool definition for this wide IP. A pool is a set of virtual servers defined and owned by a BIG-IP Controller or other host machine.
name	As part of a pool definition, defines the name of this pool. All names must be enclosed in quotation marks.
ratio	As part of a pool definition, **ratio** specifies the default weighting to use with respect to other pool types when the pool lbmode is ratio.
dynamic_ratio	Specifies whether the 3-DNS Controller treats QOS scores as ratios, and uses each server in proportion to the ratio determined by the QOS calculation. The default is **no**.

Table A.25 *Address information sub-statements*

Parameter	Description
rr_ldns	Specifies whether the 3-DNS Controller returns a list of available virtual servers available for load balancing to a client and stores the list in the browser cache. The default is **no**, which specifies that the 3-DNS Controller returns only one **A** record per query.
rr_ldns_limit	The maximum number of **A** records to return when **rr_ldns** is set to **yes**. You can enter a value between **0** and **16**. The default is **0**, which specifies that the 3-DNS Controller returns the IP addresses of all (up to 16) available virtual servers.
preferred	Specifies the load balancing mode to use for the specified pool. Each acceptable value is described in the next table. The default is **rr**.
alternate	Specifies the load balancing mode to use for the specified pool if the **preferred** mode fails. The default is **rr**. Also see the description of **default_alternate**, a **globals** sub-statement, on page A-10.
fallback	Specifies the load balancing mode to use for the specified pool if the **alternate** mode fails. If the **fallback** mode fails, the 3-DNS Controller returns the request to DNS. The default is **return_to_dns**. Also see the description of **default_fallback**, a **globals** sub-statement, on page A-10.
address	As part of a pool definition, **address** specifies the IP address of each virtual server in this pool. You can use the same virtual server in multiple pools, but not within the same pool.
port	Specifies a specific port to use for the specified virtual server. This sub-statement is optional. A port specified here overrides the wide IP's port setting. If a port is not specified here, the wide IP's port value is assumed.
ratio	As part of a virtual server's address specification, **ratio** defines the default weighting to use with respect to all virtual servers in this pool when the ratio load balancing mode is employed. The default is **1**.

Table A.25 *Address information sub-statements*

Load balancing modes

The load balancing sub-statements specify the load balancing modes to use for the wide IP in this order:

- The 3-DNS Controller attempts to load balance requests using the **preferred** mode.
- If the **preferred** mode fails, the 3-DNS Controller tries the **alternate** mode.
- If the **alternate** mode fails, the 3-DNS Controller tries the **fallback** mode.

- If the **fallback** mode fails, the request is returned to DNS. DNS attempts to resolve the request based on the contents of the zone files.

As noted in the table below, not all modes are valid for the **alternate** sub-statement. Also note that the **alternate** and **fallback** sub-statements accept two additional values, **return_to_dns** and **null**.

If you do not specify a load balancing mode, the wide IP uses the default load balancing mode defined in the **globals** statement (see page A-7).

Parameter	Description
completion_rate	Sends each new connection to the server that has the fewest number of dropped packets. Valid in a **preferred** or **fallback** sub-statement.
global_availability (**ga**)	Distributes connections to a list of servers, always sending a connection to the first available server in the list.
hops	Sends each new connection to the server that has the fewest number of network hops between the server and the client LDNS. Valid in a **preferred** or **fallback** sub-statement.
leastconn	Sends each new connection to the server that currently hosts the fewest current connections.
null	Bypasses the current load balancing method and forces the 3-DNS Controller to use the next load balancing method or, if it has cycled through all load balancing sub-statements for the pool, to the next pool. Valid in an **alternate** or **fallback** sub-statement.
packet_rate	Sends each new connection to the server that is managed by a BIG-IP Controller currently handling the least amount of network traffic (determined by the fewest number of packets currently processed by the controller).
qos	Takes these performance factors into account when determining how to distribute connections: hops, packet rate, completion rate, round trip time, and topology. You can configure how much emphasis to place on each performance factor, or you can configure the QOS mode to treat all factors as being equally important. Valid in a **preferred** or **fallback** sub-statement.
random	Distributes each new connection to a server chosen at random from the wide IP set of virtual servers.

Table A.26 *Load balancing sub-statements*

Parameter	Description
ratio	Distributes new connections across servers in proportion to a user-defined ratio.
return_to_dns	Returns the resolution request to DNS, preventing the 3-DNS Controller from using the next load balancing method or using the next available pool.
rr	Distributes connections evenly across all servers, passing each new connection to the next server in line.
rtt	Sends each new connection to the server that demonstrates the fastest round trip time between the server and the client LDNS. Valid in a **preferred** or **fallback** sub-statement.
topology	Distributes connections based on the proximity of a LDNS to a particular data center.

Table A.26 *Load balancing sub-statements*

Use the following equation to configure the QOS load balancing mode:

```
A (1/packet rate) + B (1/rtt) + C (completion rate) + D (topology)
   + E (1/hops)
```

The topology statement

The **topology** statement implements a form of wide-area IP filtering. Topology-based access control allows you to specify which data centers are acceptable for a given resolution request, based on the proximity of the data center's IP address to the requesting IP address of the LDNS server. For example, you can specify that requesting LDNS clients in North America are allowed access to data centers in North America, but not allowed access to data centers in South America.

By including a **topology** statement in your **wideip.conf** file, you can also use the topology load balancing mode, both on its own and as part of the QOS mode.

For more information and an example of a **topology** statement, see *Setting up topology-based access control*, on page 6-15.

Syntax for the topology statement

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

```
topology {
  acl_threshold   <0..4294967295>
  probe_threshold   <0..4294967295>
  limit_probes <yes |no>
  longest_match <yes | no>
    <server cidr> <LDNS cidr> <score>
}
```

Figure A.18 *Syntax for the topology statement*

Definition of topology sub-statements

Parameter	Description
acl_threshold	Specifies a score threshold. Any list record (prober IP address/LDNS IP address) score that is less than the threshold value will not have access to the listed virtual servers. You can enter a value between **0** and **4294967295**.
probe_threshold	Specifies a threshold for probing. Any list record (prober IP address/LDNS IP address) score that is less the than the probe threshold value prevents the LDNS from being probed by the specific prober. You can enter a value between **1** and **4294967295**.
limit_probes	Specifies whether to apply access control to the probing of paths. If this parameter is set to **yes**, the 3-DNS Controller requests a given BIG-IP Controller to probe only those LDNS servers that can connect to it according to the **probe_threshold** value and the topology map scores. The default is **yes**.
longest_match	In cases where there are several IP/mask items that match a particular IP address, **longest_match** specifies whether the 3-DNS Controller selects the record that is most specific, and thus has the longest mask. The default is **yes**.

Table A.27 *Topology sub-statements*

Parameter	Description
server CIDR	Specifies the server mask for a given data center. This is one of two values used to determine the longest match.
LDNS CIDR	Specifies the LDNS mask. This is one of two values used to determine the longest match.
score	Specifies the list record score, which is used for the comparison of virtual servers when the topology load balancing mode is employed.

Table A.27 *Topology sub-statements*

Probing exclusion lists

You can now create probing exclusion lists that contain a group of LDNS IP addresses whose paths the 3-DNS Controller will not probe. There are three different types of probing exclusion lists:

Syntax for the probing exclusion lists

The probing exclusion lists use the following syntax.

```
probe_acl {
  <LDNS cidr>
  <LDNS cidr>
  <LDNS cidr>
}

hops_acl {
  <LDNS cidr>
  <LDNS cidr>
  <LDNS cidr>
}

discovery_acl {
  <LDNS cidr>
  <LDNS cidr>
  <LDNS cidr>
}
```

Figure A.19 *Syntax for the probing exclusion lists*

Definition of the probing exclusion list sub-statements

Parameter	Description
probe_acl	The 3-DNS Controller restricts any **big3d** agent from probing the defined group of LDNS servers.
hops_acl	The 3-DNS Controller restricts any **big3d** agent from tracerouting the defined group of LDNS servers
discovery_acl	The 3-DNS Controller restricts any **big3d** agents from performing port discovery on the defined group of LDNS servers

Table A.28 *Probing exclusion list sub-statements*

Comments

You can insert comments anywhere you would otherwise see white space in the 3-DNS Controller configuration file.

Syntax

Note that the comment syntax depends on the environment in which you use the configuration file.

For example:

```
/* This is a 3-DNS comment as in C */
// This is a 3-DNS comment as in C++
# This is a 3-DNS comment as in common UNIX shells and Perl
```

Figure A.20 *Comment syntax*

Definition and usage

The format for comments varies by programming language; each format is described below. To avoid comment nesting problems, we recommend that you use only one comment style in your **wideip.conf** file. However, all styles may be used in a single **wideip.conf** file.

C style comments

C style comments start with the slash character, followed by the asterisk character (`/*`), and end with the asterisk character, followed with the slash character (`*/`). Because the comment is completely delimited with these characters, a comment can span multiple lines.

Note that C style comments cannot be nested. For example, the following is not valid because the entire comment ends with the first `*/`:

```
/* This is the start of a comment.
   This is still part of the comment.
/* This is an incorrect attempt to nest a comment. */
   This is no longer in any comment. */
```

Figure A.21 *Syntax for C style comments*

C++ style comments

C++ style comments start with two slash characters (`//`) and are no longer than one line in length. To have one logical comment span multiple lines, each line must start with the `//` pair.

For example:

```
// This is the start of a comment. The next line
// is a new comment line, even though it is
// logically part of the previous comment.
```

Figure A.22 *Syntax for C++ style comments*

Shell style comments

Shell style (also known as Perl style) comments start with the **#** character and are no longer than one line in length.

For example:

```
# This is the start of a comment. The next line
# is a new comment line, even though it is logically
# part of the previous comment.
```

Figure A.23 *Syntax for shell style comments*

Sample 3-DNS Controller configuration file

The following is an example of a 3-DNS Controller configuration file. Note that very few global parameters are listed. This is because you do not need to include each global parameter; you need only include those parameters for which you want to specify a value other than the default.

The following sample file contains examples of common configurations and each load balancing mode. (This sample file is several pages long.)

```
#
# Sample /etc/wideip.conf
#
# Related files are:
# /etc/named.conf
# /var/namedb/db.wip.domain.com
#


globals {
   prober       12.168.101.2     // Default prober is New York 3DNS
    encryption             yes     // Encrypt iQuery
    paths_noclobber        yes     // Don't overwrite metrics with
                                   // zeroed results
   path_ttl              2400    // Extend the life of path metrics
    rtt_probe_dynamic      yes     // Switch to tcp probing if icmp
                                   // fails
    multiplex_iq           yes     // Source port is same as dest.
                                   // port for iQuery
    use_alternate_iq_port  yes     // Use IANA registered port for
                                   // iQuery
    probe_protocol {               // Specify custom probing methods
                                   // in order desired
         icmp                     // Valid probing methods are icmp,
          dns_ver                  // udp, tcp, dns_ver, dns_dot
          dns_dot
    }
    datasize_system         90     // Memory reserved for the system
                                   // in megabytes
    datasize_reap_pct       30     // Reap percent of memory when
                                   // 3-DNS reserved memory is full
    timer_persist_cache     14400  // Dump interval of the metrics
                                   // from memory to disk
    default_iquery_protocol tcp    // Switch iQuery transport from
                                   // udp to tcp
}

include root_in              "/var/3dns/etc"
include root_out             "/var/3dns/etc"

/*** Define 3 Datacenters:  New York, Los Angeles, and Tokyo ***/

datacenter {
   name     "New York"
```

```
   3dns     192.168.101.2
   bigip    192.168.101.40
   host     192.168.105.40
}

datacenter {
   name     "Los Angeles"
   3dns     192.168.102.2
   bigip    192.168.102.40
}

datacenter {
   name     "Tokyo"
   bigip    192.168.103.40
   host     192.168.104.40
}

/*** Sync Group Definition(s) ***/

sync_group {
    name    "sync"
    3dns    192.168.101.2   // New York
    3dns    192.168.102.2   // Los Angeles
}

/*** 3DNS Definitions ***/

// New York
server {
   type            3dns
   address         192.168.101.2
   name            "3dns-newyork"
   remote {
      secure       no
      user         "root"
   }
   probe_protocol  icmp
   port            53

}

// Los Angeles
server {
   type            3dns
```

```
   address         192.168.102.2
   name            "3dns-la"
   iquery_protocol udp    //Override default iQuery transport
   remote {
      secure       no
      user         "root"
   }
   probe_protocol  tcp
   port            53
}


/*** BIG-IP Definitions ***/

// The New York BIG-IP is behind a firewall and the virtual servers
// need to be translated
server {
   type            bigip
   address         192.168.101.40
   name            "bigip-newyork"

   remote {
      secure       yes
      user         "administrator"
   }

   # Tell 3DNS about the 2 interfaces on a BIG-IP HA
   interface {
      address      192.168.101.41
      address      192.168.101.42
   }

   # Change the number of factories doing the work at big3d
   factories {
      prober     6
      discovery  1
      snmp       1
      hops       2
   }

   vs {
      address         192.168.101.50
      service         "http"
      translate {
```

```
         address      10.0.0.50
         port         80
      }
   }
   vs {
      address         192.168.101.50:25 // smtp
      translate {
         address      10.0.0.50:25
      }
   }
   vs {
      address         192.168.101.60:21 // ftp
      translate {
         address      10.0.0.60:21
      }
   }
   vs {
      address         192.168.101.60
      port            80
      translate {
         address      10.0.0.60
         port         80
      }
   }
   vs {
      address         192.168.101.70
      port            443
      translate {
         address      10.0.0.70
         port         443
      }
   }
   vs {
      address         192.168.101.70:80 // http
      translate {
         address      10.0.0.70:80
      }
   }
}

// Los Angeles
server {
   type            bigip
   address         192.168.102.40
```

```
   name            "bigip-la"
   remote {
      secure       no
   }
   vs {
      address         192.168.102.50:80
   }
   vs {
      address         192.168.102.50:25
   }
   vs {
      address         192.168.102.60:21
   }
   vs {
      address         192.168.102.60:443
   }
   vs {
      address         192.168.102.60:80
   }
   vs {
      address         192.168.102.70:80
   }
}

// Tokyo
server {
   type            bigip
   address         192.168.103.40
   name            "bigip-tokyo"
   vs {
      address         192.168.103.50:80
   }
   vs {
      address         192.168.103.50:25
   }
   vs {
      address         192.168.103.60:21
   }
   vs {
      address         192.168.103.60:80
   }
   vs {
      address         192.168.103.70:80
   }
```

```
}

/*** Host Definitions ***/

server {
   type            host
   address         192.168.104.40
   name            "host-tokyo"
   probe_protocol  icmp
   port            53
   snmp {
      agent        ucd
      community    "public"
      version      1
   }
   vs {
      address         192.168.104.50:25
   }
   vs {
      address         192.168.104.50:80
   }
   vs {
      address         192.168.104.50:443
   }
}

server {
   type            host
   address         192.168.105.40
   name            "host-la"
   probe_protocol  tcp
   prober          192.168.103.40   // Use the prober in Tokyo
   port            80
   snmp {
      agent        solstice
      community    "3dns"
      version      2
   }
   vs {
      address         192.168.105.50
      port            80
      probe_protocol  tcp
   }
   vs {
```

```
      address         192.168.105.50:25
      probe_protocol  tcp
   }
   vs {
      address         192.168.105.60:443
      probe_protocol  icmp
   }
   vs {
      address         192.168.105.60:80
      probe_protocol  icmp
   }
}

server {
   type               host
   address            192.168.106.40
   name               "host-ny"
   probe_protocol     tcp
   port               80
   iquery_protocol    udp    //Override the default iQuery transport
   snmp {
      agent           cisco-ld
      community       "public"
      version         2
   }
   vs {
      address         192.168.106.50
      port            80
      probe_protocol  tcp
   }
}

/*** WIP Definitions ***/

// Two pools with a production rule
wideip {
   address           192.168.101.50
   service           "http"
   name              "www.wip.domain.com"
   ttl               60      // increase the domain default ttl
   qos_coeff {
      rtt             21
      hops            0
      completion_rate 7
```

```
      packet_rate     5
      topology        1
   }
   pool {
      name           "Pool_1"
      ratio          2         // applies to pool_lbmode == ratio

      preferred      leastconn
      alternate      ratio

      // Pool rules can start anywhere after the pool name
      // and before the virtual servers.

      // Add some special rules to switch lbmodes:
      //   Weekday 6am-5pm: qos
      //   Weekday 5pm-6am: leastconn
      //   Weekend        : packet_rate

      /*** If a weekday ***/
      rule "myRule1"
      if(day != "sat" && day != "sun") {

         /*** If during business hours 6am-5pm, do QOS ***/
         rule "myRule2"
         if(preferred != "qos" && (time >= "6:00" && time <=
"17:00")) {

            // switch the lbmode and log a message that it happened
            preferred qos
            log("Switching preferred to $preferred")
         }

         /*** Otherwise, do least connections ***/
         else {
            rule "myRule3"
            if(preferred != "leastconn") {
               preferred leastconn
               log("Switching preferred to $preferred")
            }
         }
      }

      /*** If weekend, switch to packet rate ***/
      else {
```

```
         rule "weekendPolicy"
         if(preferred != "packet_rate") {
              preferred packet_rate
              log("Switching preferred to $preferred")
         }
      }

      address       192.168.101.50   ratio 2
      address       192.168.102.50   ratio 1
      address       192.168.103.50   ratio 1

   }
   pool {
      name           "Pool_2"
      ratio          1
      preferred      rr
      address       192.168.102.60   ratio 2
      address       192.168.103.60   ratio 1
   }
}

// Global availability
wideip {
   address           192.168.101.60
   port              80 // http
   name              "cgi.wip.domain.com"
   pool {
      name           "mypool"
      preferred      ga
      address       192.168.101.60
      address       192.168.102.60
      address       192.168.103.60
   }
}


// Round robin pool load balancing between bigip and hosts
// This site runs a catalog and shopping cart and only wishes
// to send client to a datacenter if services are up on both
// ports 80 and 443.
wideip {
   address           192.168.101.70
   port              80 // http
   port_list         80  443          // e-commerce
```

```
   name              "ssl.wip.domain.com"
   pool_lbmode       rr
   pool {
      name           "bigip_pool"
      ratio          2
      preferred      qos
      alternate      ratio
      address       192.168.101.70   ratio 7
      address       192.168.102.60   ratio 2
   }
   pool {
      name           "host_pool"
      ratio          1
      preferred      ratio
      address       192.168.104.50   ratio 2
      address       192.168.105.60   ratio 1
   }
}

// Mixing hosts and BIG-IP virtual servers
// Ratio pool load balancing between bigip and hosts
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
   }
}
```

```
// Least connections with ratio as an alternate
wideip {
   address           192.168.102.60
   service           "ftp"
   name              "ftp.wip.domain.com"
   pool {
      name           "main_pool"
      preferred      leastconn
      alternate      ratio
      address       192.168.101.60   ratio 2
      address       192.168.102.60   ratio 4
      address       192.168.103.60
   }
}

// Preferred set to QOS with attributes dynamic ratio and ldns_rr
// Alternate set to leastconn
wideip {
   address           192.168.103.70:80
   name              "www2.wip.domain.com"
   pool {
      name           "pool_1"
      dynamic_ratio  yes
      rr_ldns        yes
      rr_ldns_limit  10 // Return up to 10 available virtual servers
      preferred      qos
      alternate     leastconn
      address       192.168.101.60
      address       192.168.102.60
      address       192.168.103.70
   }
}


// Global availability  pool load balancing between bigip
datacenters
// with specialized use of preferred, alternate, and fallback load
// balancing methods null and return_to_dns.
wideip {
   address           192.168.102.70
   port              80
   name              "www.domain.com"
   alias             "home.domain.com"
```

```
   ttl               120
   pool_lbmode       ga
   pool {
      name           "New York"
      ratio          2
      preferred      leastconn
      alternate      null
      fallback       null          // null here allows fail over to
next pool
      address        192.168.101.50
      address        192.168.101.60
      address        192.168.101.70
   }
   pool {
      name           "Los Angeles"
      ratio          1
      preferred      leastconn
      alternate      null
      fallback       null  // null here allows fail over to next pool
      address        192.168.102.50
      address        192.168.102.60
      address        192.168.102.70
   }
   pool {
      name           "Tokyo"
      ratio          1
      preferred      leastconn
      alternate      null
#     fallback       return_to_dns // (this is the fallback default)
      address        192.168.103.50
      address        192.168.103.60
      address        192.168.103.70
   }
}

// Round trip time load balancing with topology as alternate load
// balancing (see topology below)
wideip {
   address           192.168.103.60
   port              80
   name              "ntp.wip.domain.com"
   pool {
      name           "poolA"
      preferred      rtt
```

```
      alternate      topology
      address        192.168.101.60   // New York
      address        192.168.102.60   // Los Angeles
      address        192.168.103.60   // Tokyo
   }
}

//
//
wideip {
   address          192.168.103.60
   port             80
   name             "my.wip.domain.com"
   ttl              0                  //A REC ttl
   persist          yes                //Persist on LDNS IP to
                                       // virtual server
   persist_ttl      1800               // How long to persist
   pool_lbmode      rr
   pool {
      name          "Pool One"
      ttl           10                 // Override A REC ttls
                                       // per pool
      preferred     rtt
      alternate     topology
      address       192.168.101.50     // New York
      address       192.168.102.50     // Los Angeles
      address       192.168.103.50     // Tokyo
   }
   pool {
      name          "Pool Two"
      ttl           30
      preferred     vs_capacity        // select the virtual server
                                      // that has the most nodes up
      alternate     topology
      address       192.168.101.60     // New York
      address       192.168.102.60     // Los Angeles
      address       192.168.103.60     // Tokyo
   }
}

/*** Topological distribution and access control ***/

topology {
   acl_threshold 5
```

```
   probe_threshold 5
   limit_probes  yes
   longest_match yes

// server/mask   ldns/mask   score


//////////////////////////
// North American LDNS's:
//   198.0.0.0/8
//   199.0.0.0/8

// North America Priority List
//
// 1. New York
// 2. L.A.
// 3. Tokyo

   // New York
   192.168.101.0/24   198.0.0.0/8   30
   192.168.101.0/24   199.0.0.0/8   30

   // Los Angeles
   192.168.102.0/24   198.0.0.0/8   20
   192.168.102.0/24   199.0.0.0/8   20

   // Tokyo
   192.168.103.0/24   198.0.0.0/8   10
   192.168.103.0/24   199.0.0.0/8   10


//////////////////////////
// South American LDNS's:
//   200.0.0.0/8
//   201.0.0.0/8

// South America Priority List
//
// 1. Tokyo
// 2. L.A.
// (New York excluded by acl_threshold)

   // Tokyo
   192.168.103.0/24   200.0.0.0/8   30
```

```
   192.168.103.0/24   201.0.0.0/8   30

   // Los Angeles
   192.168.102.0/24   200.0.0.0/8   20
   192.168.102.0/24   201.0.0.0/8   20

   // New York
   192.168.101.0/24   200.0.0.0/8   0
   192.168.101.0/24   201.0.0.0/8   0

//////////////////////////
// Wildcard List Record
//
// By default, if a list record is not found in the
// topology map for an LDNS, the score is assumed to
// be 0. By including the following "wildcard" list
// record, all other LDNS's (not North or South America
// as specified above) are assigned a score of 1 so
// the acl_threshold does not indicate that the
// virtual servers are down.

   0.0.0.0/0   0.0.0.0/0   1


}



/*
 *Probing exclude files
 */

// probe_acl - The 3-DNS Controller will not have any
// big3d agents probe this group of LDNS servers.

probe_acl {
     10.20.30.0/24
     192.168.0.0/16
}

// hops_acl - The 3-DNS Controller will not have any
// big3d agents traceroute this group of LDNS servers.

hops_acl {
```

```
     10.20.30.0/24
     192.168.0.0/16
}

// discovery_acl - The 3-DNS Controller will not have any big3d
// agents perform port discovery on this group of LDNS servers.

discovery_acl {
     10.20.30.0/24
     192.168.0.0/16
}



/*
 * Some global production rules
 */

rule "everyExample"
every(60) {
   log("We should do something nifty every minute")
}


rule "whenExample"
when(SIGHUP) {
   log("System was just restarted (i.e. ndc reload)")
}
```

Sample 3-DNS Controller configuration file using include files

The following examples show the same sample configuration as shown in the previous section, but with the incorporation of include files.

Sample wideip.conf using include files

```
#
# Sample /etc/wideip.conf using include files
#
include root_in              "/var/3dns/etc"
include root_out             "/var/3dns/etc"
include global               "globals.conf"
include datacenter           "datacenters.conf"
include sync_group           "sync_groups.conf"
include server               "3dns.conf"
include server               "bigips.conf"
include server               "hosts.conf"
include wideip               "wideips.conf"
include 3dscript             "prodrules.conf"
include topology             "topology.conf"
```

Sample globals.conf

```
// globals.conf
globals {
   prober        192.168.101.2    // Default prober is New York 3DNS
   encryption    yes               // Encrypt iQuery
   paths_noclobber yes         // Don't overwrite metrics with
                                   // zeroed results
   path_ttl     400              // Extend the life of path metrics
   rtt_probe_dynamic yes       // Switch to tcp probing if icmp
                                   // fails
   multiplex_iq  yes             // Source port is same as dest.
                                   // port for iQuery
   use_alternate_iq_port yes       // Use IANA registered port for
                                   // iQuery
}
```

Sample datacenters.conf

```
// datacenters.conf
/*** Define 3 Datacenters:  New York, Los Angeles, and Tokyo ***/
```

```
datacenter {
   name     "New York"
   3dns     192.168.101.2
   bigip    192.168.101.40
   host     192.168.105.40
}
datacenter {
   name     "Los Angeles"
   3dns     192.168.102.2
   bigip    192.168.102.40
}
datacenter {
   name     "Tokyo"
   bigip    192.168.103.40
   host     192.168.104.40
}
```

Sample sync_groups.conf

```
// sync_groups.conf
sync_group {
    name    "sync"
    3dns    192.168.101.2   // New York
    3dns    192.168.102.2   // Los Angeles
}
```

Sample 3dns.conf

```
// 3dns.conf
// New York
server {
   type            3dns
   address         192.168.101.2
   name            "3dns-newyork"
   remote {
      secure       no
      user         "root"
   }
   probe_protocol  icmp
   port            53
}
// Los Angeles
server {
   type            3dns
   address         192.168.102.2
   name            "3dns-la"
```

```
   remote {
      secure       no
      user         "root"
   }
   probe_protocol  tcp
   port            53
}
```

Sample bigips.conf

```
bigips.conf
// The New York BIG-IP is behind a firewall and the virtual servers
// need to be translated
server {
   type            bigip
   address         192.168.101.40
   name            "bigip-newyork"
   remote {
      secure       yes
      user         "administrator"
   }
   # Tell 3-DNS about the 2 interfaces on a BIG-IP HA
   interface {
      address      192.168.101.41
      address      192.168.101.42
   }
   # Change the number of factories doing the work at big3d
   factories {
      prober     6
      discovery  1
      snmp       1
      hops       2
   }
include vs "bigip_ny/vs.conf"
include path "bigip_ny/path.conf"
}
// Los Angeles
server {
   type            bigip
   address         192.168.102.40
   name            "bigip-la"
   remote {
      secure       no
   }
include vs "bigip_la/vs.conf"
```

```
include path "bigip_la/path.conf"
}
// Tokyo
server {
   type            bigip
   address         192.168.103.40
   name            "bigip-tokyo"
   vs {
      address         192.168.103.50:80
   }
include vs "bigip_tokyo/vs.conf"
include path "bigip_tokyo/path.conf"
}
```

Sample bigip_ny/vs.conf

```
// bigip_ny/vs.conf
   vs {
      address         192.168.101.50
      service         "http"
      translate {
         address      10.0.0.50
         port         80
      }
   }
   vs {
      address         192.168.101.50:25 // smtp
      translate {
         address      10.0.0.50:25
      }
   }
   vs {
      address         192.168.101.60:21 // ftp
      translate {
         address      10.0.0.60:21
      }
   }
   vs {
      address         192.168.101.60
      port            80
      translate {
         address      10.0.0.60
         port         80
      }
```

```
   }
   vs {
      address          192.168.101.70
      port             443
      translate {
         address       10.0.0.70
         port          443
      }
   }
   vs {
      address          192.168.101.70:80 // http
      translate {
         address       10.0.0.70:80
      }
   }
```

Sample bigip_la/vs.conf

```
// bigip_la/vs.conf
   vs {
      address          192.168.102.50:80
   }
   vs {
      address          192.168.102.50:25
   }
   vs {
      address          192.168.102.60:21
   }
   vs {
      address          192.168.102.60:443
   }
   vs {
      address          192.168.102.60:80
   }
   vs {
      address          192.168.102.70:80
   }
```

Sample bigip_tokyo/vs.conf

```
// bigip_tokyo/vs.conf
   vs {
      address          192.168.103.50:25
   }
   vs {
      address          192.168.103.60:21
```

```
   }
   vs {
      address         192.168.103.60:80
   }
   vs {
      address         192.168.103.70:80
   }
```

Sample hosts.conf

```
// hosts.conf
server {
   type            host
   address         192.168.104.40
   name            "host-tokyo"
   probe_protocol  icmp
   port            53
   snmp {
      agent        ucd
      community    "public"
      version      1
   }
include vs "host_tokyo/vs.conf"
include path "host_tokyo/path.conf"
}
server {
   type            host
   address         192.168.105.40
   name            "host-la"
   probe_protocol  tcp
   prober          192.168.103.40   // Use the prober in Tokyo
   port            80
   snmp {
      agent        solstice
      community    "3dns"
      version      2
   }
include vs "host_la/vs.conf"
include path "host_la/path.conf"
}
```

Sample host_tokyo/vs.conf

```
// host_tokyo/vs.conf
   vs {
      address         192.168.104.50:25
```

```
   }
   vs {
      address         192.168.104.50:80
   }
   vs {
      address         192.168.104.50:443
   }
```

Sample host_la/vs.conf

```
// host_la/vs.conf
   vs {
      address         192.168.105.50
      port            80
      probe_protocol  tcp
   }
   vs {
      address         192.168.105.50:25
      probe_protocol  tcp
   }
   vs {
      address         192.168.105.60:443
      probe_protocol  icmp
   }
   vs {
      address         192.168.105.60:80
      probe_protocol  icmp
   }
```

Sample wideips.conf

```
// wideips.conf
// Two pools with a production rule
wideip {
   address           192.168.101.50
   service           "http"
   name              "www.wip.domain.com"
   ttl               60      // increase the domain default ttl
   qos_coeff {
      rtt             21
      hops            0
      completion_rate 7
      packet_rate     5
      topology        1
   }
   pool {
```

```
      name           "Pool_1"
      ratio          2         // applies to pool_lbmode == ratio
      preferred      leastconn
      alternate      ratio
      address       192.168.101.50   ratio 2
      address       192.168.102.50   ratio 1
      address       192.168.103.50   ratio 1
   }
   pool {
      name           "Pool_2"
      ratio          1
      preferred      rr
      address       192.168.102.60   ratio 2
      address       192.168.103.60   ratio 1
   }
}

// Global availability
wideip {
   address           192.168.101.60
   port              80 // http
   name              "cgi.wip.domain.com"
   pool {
      name           "mypool"
      preferred      ga
      address       192.168.101.60
      address       192.168.102.60
      address       192.168.103.60
   }
}

// Round robin pool load balancing between bigip and hosts
// This site runs a catalog and shopping cart and only wishes
// to send client to a datacenter if services are up on both
// ports 80 and 443.
wideip {
   address           192.168.101.70
   port              80 // http
   port_list         80  443          // e-commerce
   name              "ssl.wip.domain.com"
   pool_lbmode       rr
   pool {
      name           "bigip_pool"
      ratio          2
```

```
      preferred      qos
      alternate      ratio
      address       192.168.101.70   ratio 7
      address       192.168.102.60   ratio 2
   }
   pool {
      name           "host_pool"
      ratio          1
      preferred      ratio
      address       192.168.104.50   ratio 2
      address       192.168.105.60   ratio 1
   }
}

// Mixing hosts and BIG-IP virtual servers
// Ratio pool load balancing between bigip and hosts
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
   }
}

// Least connections with ratio as an alternate
wideip {
   address           192.168.102.60
   service           "ftp"
```

```
   name              "ftp.wip.domain.com"
   pool {
      name           "main_pool"
      preferred      leastconn
      alternate      ratio
      address       192.168.101.60   ratio 2
      address       192.168.102.60   ratio 4
      address       192.168.103.60
   }
}

// Preferred set to QOS with attributes dynamic ratio and ldns_rr
// Alternate set to leastconn
wideip {
   address           192.168.103.70:80
   name              "www2.wip.domain.com"
   pool {
      name           "pool_1"
      dynamic_ratio  yes
      rr_ldns        yes
      rr_ldns_limit  10 // Return up to 10 available virtual servers
      preferred      qos
      alternate     leastconn
      address       192.168.101.60
      address       192.168.102.60
      address       192.168.103.70
   }
}

// Global availability pool load balancing between bigip
// datacenters with specialized use of preferred, alternate, and
// fallback load balancing methods null and return_to_dns.
wideip {
   address           192.168.102.70
   port              80
   name              "www.domain.com"
   alias             "home.domain.com"
   ttl               120
   pool_lbmode       ga
   pool {
      name           "New York"
      ratio          2
      preferred      leastconn
      alternate      null
```

```
      fallback       null  // null here allows fail over to next pool
       address       192.168.101.50
       address       192.168.101.60
       address       192.168.101.70
   }
   pool {
       name           "Los Angeles"
       ratio          1
       preferred      leastconn
       alternate      null
      fallback       null  // null here allows fail over to next pool
       address       192.168.102.50
       address       192.168.102.60
       address       192.168.102.70
   }
   pool {
       name           "Tokyo"
       ratio          1
       preferred      leastconn
       alternate      null
#     fallback       return_to_dns // (this is the fallback default)
       address       192.168.103.50
       address       192.168.103.60
       address       192.168.103.70
   }
}

// Round trip time load balancing with topology as alternate load
// balancing (see topology below)
wideip {
   address           192.168.103.60
   port              80
   name              "ntp.wip.domain.com"
   pool {
       name          "poolA"
       preferred     rtt
       alternate     topology
       address       192.168.101.60   // New York
       address       192.168.102.60   // Los Angeles
       address       192.168.103.60   // Tokyo
   }
}
```

Sample prodrules.conf

```
// prodrules.conf
   // Pool rules can start anywhere after the pool name
      // and before the virtual servers.

      // Add some special rules to switch lbmodes:
      //   Weekday 6am-5pm: qos
      //   Weekday 5pm-6am: leastconn
      //   Weekend        : packet_rate

      /*** If a weekday ***/
      rule "myRule1"
      if(day != "sat" && day != "sun") {

         /*** If during business hours 6am-5pm, do QOS ***/
         rule "myRule2"
         if(preferred != "qos" && (time >= "6:00" && time <=
"17:00")) {
            // switch the lbmode and log a message that it happened
            preferred qos
            log("Switching preferred to $preferred")
         }

         /*** Otherwise, do least connections ***/
         else {
            rule "myRule3"
            if(preferred != "leastconn") {
               preferred leastconn
               log("Switching preferred to $preferred")
            }
         }
      }

      /*** If weekend, switch to packet rate ***/
      else {
         rule "weekendPolicy"
         if(preferred != "packet_rate") {
              preferred packet_rate
              log("Switching preferred to $preferred")
         }
      }
```

Sample topology.conf

```
// topology.conf
topology {
   acl_threshold 5
   probe_threshold 5
   limit_probes  yes
   longest_match yes

// server/mask   ldns/mask   score


//////////////////////////
// North American LDNS's:
//   198.0.0.0/8
//   199.0.0.0/8

// North America Priority List
//
// 1. New York
// 2. L.A.
// 3. Tokyo

   // New York
   192.168.101.0/24   198.0.0.0/8   30
   192.168.101.0/24   199.0.0.0/8   30

   // Los Angeles
   192.168.102.0/24   198.0.0.0/8   20
   192.168.102.0/24   199.0.0.0/8   20

   // Tokyo
   192.168.103.0/24   198.0.0.0/8   10
   192.168.103.0/24   199.0.0.0/8   10


//////////////////////////
// South American LDNS's:
//   200.0.0.0/8
//   201.0.0.0/8

// South America Priority List
//
// 1. Tokyo
// 2. L.A.
```

```
// (New York excluded by acl_threshold)

   // Tokyo
   192.168.103.0/24   200.0.0.0/8   30
   192.168.103.0/24   201.0.0.0/8   30

   // Los Angeles
   192.168.102.0/24   200.0.0.0/8   20
   192.168.102.0/24   201.0.0.0/8   20

   // New York
   192.168.101.0/24   200.0.0.0/8   0
   192.168.101.0/24   201.0.0.0/8   0

//////////////////////////
// Wildcard List Record
//
// By default, if a list record is not found in the
// topology map for an LDNS, the score is assumed to
// be 0. By including the following "wildcard" list
// record, all other LDNS's (not North or South America
// as specified above) are assigned a score of 1 so
// the acl_threshold does not indicate that the
// virtual servers are down.

   0.0.0.0/0   0.0.0.0/0   1


}


/*
 * Some global production rules
 */

rule "everyExample"
every(60) {
   log("We should do something nifty every minute")
}


rule "whenExample"
when(SIGHUP) {
   log("System was just restarted (i.e. ndc reload)")
```

```
}
```

Understanding cur_ values

You may notice several **cur_** values in your **wideip.conf** file. The purpose of **cur_** values is to pre-load the database with previously collected statistics and metrics. The collected statistics and metrics are useful if you want to quickly restart a 3-DNS Controller without a temporary loss of intelligence.

Do not edit these statements unless you are a very experienced 3-DNS Controller user, or you are instructed to do so by your vendor.

How cur_ values are used

You may notice **cur_** values in **server**, **vs**, **path**, or **wideip** definitions. Examples for each type of definition follow.

Example: server definition

```
// New York BIG-IP Controller
server {
  type bigip
  address 192.168.101.40
  cur_packet_rate 139
  cur_ok 1
  [virtual server definitions]
}
```

Figure A.24 *cur_ values in a server definition*

In the above example, the **cur_** values indicate the following.

Parameter	Description
cur_packet_rate	Indicates the number of packets per second sent during the last sample period.
cur_ok	Indicates the state of the specified server. The options are: **1 (Up)**, **2 (Down)**, **3 (Waiting)**, **4 (Alert)**, and **5 (Panic)**.

Table A.29 *cur_values in a server definition*

Example: vs definition

```
vs {
  address 192.168.102.50:80
  cur_serv_cnt 1
  cur_connections 0
  cur_picks 39
  cur_refreshes 783
}
```

Figure A.25 *cur_ values in a virtual server definition*

In the above example, the **cur_** values indicate the following:

Parameter	Description
cur_nodes_up	Indicates the number of active servers serving the specified virtual server.
cur_connections	Indicates the number of connections to the specified virtual server.
cur_picks	Indicates the number of times the specified virtual server was returned by the 3-DNS Controller.
cur_refreshes	Indicates the number of times the server and connection counts were refreshed with new data.

Table A.30 *cur_ values in a virtual server definition*

Example: path definition

```
path {
  address 10.25.50.100 // LDNS
  cur_rtt 102382
  cur_completion_rate 10000
  cur_picks 239
  cur_accesses 302
}
```

Figure A.26 *cur_ values in a path definition*

In the above example, the **cur_** values indicate the following:

Parameter	Description
cur_rtt	Indicates the **round trip time (RTT)**, which is a calculation of the time (in microseconds) that the specified machine takes to respond to a probe issued by the 3-DNS Controller.
cur_completion_rate	Indicates the percentage of completed packets versus lost packets, using this equation: [**1** - (packets received / sent)] X **10000**.
cur_picks	Indicates the number of times this path's data resulted in the virtual server being chosen for a connection. This only applies if a wide IP is doing dynamic load balancing (using path data).
cur_accesses	Indicates the number of times this path was considered when performing dynamic load balancing.

Table A.31 *cur_ values in a path definition*

Example: wide IP definition

```
wideip {
  address 192.168.102.70
  name "www.domain.com"
  port 80
  cur_preferred 143982
  cur_alternate 108090
  cur_fallback 130094
  cur_returned_to_dns 23872
  [virtual server definitions]
}
```

Figure A.27 *cur_ values in a wide IP definition*

In the above example, the **cur_** values indicate the following:

Parameter	Description
cur_preferred	Indicates the number of times the specified wide IP was resolved by the **preferred** load balancing mode.
cur_alternate	Indicates the number of times the specified wide IP was resolved by the **alternate** load balancing mode.
cur_fallback	Indicates the number of times the specified wide IP was resolved by the **fallback** load balancing mode.
cur_returned_to_dns	Indicates the number of times the specified wide IP did not find a suitable virtual server to return using the **preferred**, **alternate**, or **fallback** load balancing modes. In this situation, the 3-DNS Controller returns the wide IP key (fallback address) as specified in the zone file.

Table A.32 *cur_ values in a wide IP definition*

◆ Tip

To find out how many times the 3-DNS Controller received resolution requests for this wide IP, add the values for ***cur_preferred****,* ***cur_alternate****, and* ***cur_fallback****.*

F5 Networks

3·DNS®

Distributed Traffic Controller

B

3-DNS Controller Utilities and Scripts

Using utilities and scripts

The 3-DNS Controller includes several utilities and scripts. These utilities and scripts allow you to configure the DNS and the various features of the 3-DNS Controller.

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 via the Configuration utility

1. Log in to the Configuration utility.
2. From the On-line 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 via 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
```

Scripts

This section 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, which is discussed in the section titled *Working with the 3-DNS Maintenance menu*, on page 7-1.

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 into 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 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, 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. **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. This script is useful for 3-DNS Controller updates.

big3d_install performs the following procedure on each BIG-IP Controller:

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. The matrix file is a file that lists version numbers for all BIG-IP Controllers known to the 3-DNS Controller and the version numbers of the **big3d** agent and **named** utility running on each BIG-IP Controller.
3. Adds the following to the bottom of the **/etc/rc.local** file:

```
if [ -f /usr/sbin/big3d ]; then
  echo -n "big3d":  /usr/sbin/big3d 2>
/dev/null
fi
```

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`

The archived files are restored 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`

The archived files are restored 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.

F5 Networks

3·DNS®

Distributed Traffic Controller

C

BIND 8 Information

BIND 8 overview

The 3-DNS Controller version 2.1 is based on BIND 8.2.2, patch level 5.

For more information on BIND, refer to the Internet Software Consortium Web site at **www.isc.org**.

BIND 8 has the advantage of being more configurable than earlier versions of BIND. New areas of configuration, such as access control lists (ACLs) and categorized logging, are now available. You can selectively apply more options, rather than being required to apply options to all zones. To incorporate this new technology and provide for future enhancements, BIND 8 requires a new format for configuration files.

A BIND 8 configuration file consists of two types of information: statements and comments. Both of these are described in the following sections. This appendix also describes the relationship between BIND and 3-DNS Controller wide IP definitions.

Statements

BIND statements end with a semicolon. Statements can contain blocks of sub-statements, which are also terminated with a semicolon.

The following statements are supported:

Statement	Description
acl	Defines a named IP address matching list, for access control and other uses.
include	Includes a file.
key	Specifies key information for use in authentication and authorization.
logging	Specifies what the server logs are, and where the log messages are sent. This statement may only be used once per configuration.

Table C.1 *BIND statements*

Statement	Description
options	Controls global server configuration options and sets defaults for other statements. This statement may only be used once per configuration.
server	Sets certain configuration options on a per-server basis.
zone	Defines a zone.

Table C.1 *BIND statements*

acl statement

The **acl** statement creates a named address match list. It gets its name from a primary use of address match lists: Access Control Lists (ACLs).

Note that an address match list's name must be defined with **acl** before it can be used elsewhere; no forward references are allowed.

The following ACLs are built in:

ACL	Description
any	Allows all hosts.
none	Denies all hosts.
localhost	Allows the IP addresses of all interfaces on the system.
localnets	Allows any host on a network for which the system has an interface.

Table C.2 *Supported ACLs*

Syntax

```
acl <name> {
  address_match_list
};
```

key statement

The **key** statement defines a key ID which can be used in a server statement to associate an authentication method with a particular name server.

The **key** statement is intended for future use by the server. It is checked for syntax but is otherwise ignored.

Syntax

```
key <key_id>{
  algorithm <algorithm_id>;
  secret <secret_string>;
};
```

logging statement

The **logging** statement configures a wide variety of logging options for the name server.

Syntax

```
logging {
  [ channel <channel_name> {
    ( file <path_name>
       [ versions ( number | unlimited ) ]
       [ size <size_spec>  ]
     | syslog ( kern | user | mail | daemon |
       auth | syslog | lpr |
       news | uucp | cron | authpriv | ftp |
       local0 | local1 | local2 | local3 |
       local4 | local5 | local6 | local7 )
     | null );
    [ severity ( critical | error | warning|notice
    |
```

```
                info  | debug [ level ] | dynamic ); ]
    [ print-category <yes | no>; ]
    [ print-severity <yes | no>; ]
    [ print-time <yes | no>; ]
  }; ]
  [ category <category_name> {
    <channel_name>; [ <channel_name>; ... ]
  }; ]
  ...
};
```

options statement

The **options** statement sets up global options to be used by BIND. This statement should appear only once in a configuration file; if BIND finds more than one occurrence, BIND honors the first. When this happens, BIND generates a warning alerting you that your configuration contains multiple **options** statements. If BIND does not find an **options** statement in the configuration file, BIND uses an **options** block, with each option set to its default.

Syntax

```
options {
  [ directory <path_name>; ]
  [ named-xfer <path_name>; ]
  [ dump-file <path_name>; ]
  [ memstatistics-file <path_name>; ]
  [ pid-file <path_name>; ]
  [ statistics-file <path_name>; ]
  [ auth-nxdomain <yes | no>; ]
  [ deallocate-on-exit <yes | no>; ]
  [ fake-iquery <yes | no>; ]
  [ fetch-glue <yes | no>; ]
  [ host-statistics <yes | no>; ]
  [ multiple-cnames <yes | no>; ]
  [ notify <yes | no>; ]
  [ recursion <yes | no>; ]
  [ forward ( only | first ); ]
  [ forwarders { [<in_addr>;[<in_addr>;...]]};]
  [ check-names (master | slave | response )
   ( warn | fail | ignore); ]
  [ allow-query { <address_match_list> }; ]
  [ allow-transfer { <address_match_list> }; ]
  [ listen-on [ port <ip_port> ]
   { <address_match_list> }; ]
```

```
  [ query-source [ address ( <ip_addr> | * ) ]
  [ port ( <ip_port> | * ) ] ; ]
  [ max-transfer-time-in <number>; ]
  [ transfer-format (one-answer|many-answers);]
  [ transfers-in <number>; ]
  [ transfers-out <number>; ]
  [ transfers-per-ns <number>; ]
  [ coresize <size_spec> ; ]
  [ datasize <size_spec> ; ]
  [ files <size_spec> ; ]
  [ stacksize <size_spec> ; ]
  [ cleaning-interval <number>; ]
  [ interface-interval <number>; ]
  [ statistics-interval <number>; ]
  [ topology { <address_match_list> }; ]
};
```

server statement

The **server** statement defines the characteristics associated with a remote name server.

Syntax

```
server <ip_addr> {
  [ bogus <yes | no>; ]
  [ transfers <number>; ]
  [ transfer-format (one-answer|many-answers);]
  [ keys { <key_id> [key_id ... ] }; ]
};
```

zone statement

The **zone** statement defines a zone.

Syntax

```
zone <domain_name> [ ( in|hs|hesiod|chaos )]{
  type master;
  file <path_name>;
  [ check-names ( warn | fail | ignore ); ]
  [ allow-update { <address_match_list> }; ]
  [ allow-query { <address_match_list> }; ]
  [ allow-transfer { <address_match_list> }; ]
  [ notify <yes | no>; ]
  [ also-notify { <ip_addr>; [ <ip_addr>;...]};
};
zone <domain_name> [ ( in|hs|hesiod|chaos) ] {
  type ( slave | stub );
  [ file <path_name>; ]
  masters { <ip_addr>; [ <ip_addr>; ... ] };
  [ check-names ( warn | fail | ignore ); ]
  [ allow-update { <address_match_list> }; ]
  [ allow-query { <address_match_list> }; ]
  [ allow-transfer { <address_match_list> }; ]
  [ max-transfer-time-in <number>; ]
  [ notify <yes | no>; ]
  [ also-notify { <ip_addr>; [ <ip_addr>;...]};
};
zone "." [ ( in | hs | hesiod | chaos ) ] {
  type hint;
  file <path_name>;
  [ check-names ( warn | fail | ignore ); ]
};
```

Comments

BIND 8 comments follow syntax rules that are similar to the 3-DNS Controller comments syntax rules.

You can insert comments anywhere you would otherwise see white space in a BIND configuration file.

Syntax

Note that the comment syntax depends on the environment in which you use the configuration file. For example:

```
/* This is a BIND comment as in C */
// This is a BIND comment as in C++
# This is a BIND comment as in common UNIX shells
   and Perl
```

Definition and usage

The format for comments varies by programming language; each format is described below.

C style comments

C style comments start with the slash character, followed by the asterisk character (**/***), and end with the asterisk character, followed with the slash character (***/**). Because the comment is completely delimited with these characters, a comment can span multiple lines.

Note that C style comments cannot be nested. For example, the following is not valid because the entire comment ends with the first ***/**:

```
/* This is the start of a comment.
   This is still part of the comment.
/* This is an incorrect attempt to nest a comment. */
   This is no longer in any comment. */
```

C++ style comments

C++ style comments start with two slash characters (//) and are no longer than one line in length. To have one logical comment span multiple lines, each line must start with the // pair.

For example:

```
// This is the start of a comment. The next line
// is a new comment line, even though it is
// logically part of the previous comment.
```

Shell style comments

Shell style (also known as Perl style) comments start with the # character and are no longer than one line in length.

For example:

```
# This is the start of a comment. The next line
# is a new comment line, even though it is logically
# part of the previous comment.
```

WARNING

You cannot use the semicolon (;) character to start a comment as you would in a zone file. The semicolon indicates the end of a configuration statement. Text following a semicolon is interpreted as the start of the next statement.

Converting older configuration files to BIND 8 format

You can convert BIND 4.9.x configuration files to the BIND 8 format using **/etc/named-bootconf.pl**, a Perl script that is part of the BIND 8.1 source kit.

Relating BIND information to 3-DNS Controller wide IP definitions

We recommend that you use NameSurfer, which is included with the 3-DNS Controller, to configure BIND. However, if you want to manually configure BIND and understand how BIND relates to wide IP definitions, this section provides an example procedure for adding a wide IP when BIND is manually configured.

◆ Note

*The following information assumes you have read O'Reilly & Associates' book **DNS and BIND** (third edition). You can purchase this book from a technical bookstore.*

Before defining a wide IP

Before configuring a wide IP, gather your 3-DNS Controller, BIG-IP Controller, host, and virtual server information.

The following information includes sample wide IP statements that derive from the sample configuration file that starts on page A-48. The sample wide IP statement configures a wide IP for the **domain.com** domain, where the IP addresses assigned to the 3-DNS Controller interfaces are shown in the following table.

3-DNS Controller	IP address
New York	192.168.101.2
Los Angeles	192.168.102.2

Table C.3 *3-DNS Controller addresses*

Gather your BIG-IP Controller and host configuration information so you can easily see which virtual servers have the replicated content.

For example, create tables like the following. In the first table, list each server.

Server	IP address	Server type
New York	192.168.101.40	BIG-IP Controller
Los Angeles	192.168.102.40	BIG-IP Controller
Tokyo	192.168.103.40	BIG-IP Controller
Tokyo	192.168.104.40	Host
Los Angeles	192.168.105.40	Host

Table C.4 *Server addresses*

Next, create a table that lists the virtual servers managed by each BIG-IP Controller (include only those that host content for the domain you are load balancing). For example, each virtual server in the following table is owned by a different BIG-IP Controller, yet each contains identical content.

BIG-IP Controller	Virtual server	Virtual port
New York	192.168.101.50	80
Los Angeles	192.168.102.50	80
Tokyo	192.168.103.50	80

Table C.5 *Virtual servers owned by BIG-IP Controllers*

You configure virtual servers as part of the BIG-IP Controller configuration process. See the ***BIG-IP Controller Administrator Guide.*** List the other host machines and the IP addresses of the virtual servers that contain the same content.

Host	Virtual server	Virtual port
Tokyo	192.168.104.50	80
Los Angeles	192.168.105.50	80

Table C.6 *Virtual servers owned by host machines*

Defining a wide IP

The following procedure shows how to manually add a wide IP.

To add a wide IP and manually configure BIND

1. Find or create the top level domain configuration file. This file is usually found in the **/etc** directory.
   - For BIND 4, type the following line in the **named.boot** file:

     ```
     primary domain.com db.domain.com
     ```

- For BIND 8, type the following in the **named.conf** file:

```
zone "domain.com" IN {
  type master;
  file "db.domain.com";
};
```

To specify a type other than **master**, see the syntax for the **zone** statement on page C-7.

2. If your network's master DNS is not a 3-DNS Controller, create a new subdomain to be controlled by the 3-DNS Controller.

   For example, to create a subdomain called **wip.domain.com**, do one of the following:

   - Add the new subdomain to the **named.conf** file with the following lines:

```
zone "wip.domain.com" IN {
  type master;
  file "db.wip.domain.com";
};
```

Figure C.1 *Sample code to add subdomain*

- If the 3-DNS Controller does not manage the top level domain, the subdomain must be delegated to each 3-DNS Controller on your network. To delegate the

domain to each 3-DNS Controller in your network, add lines like the following to the top-level domain database file (**db.domain.com** in this example):

```
wip IN NS 3dns.newyork
    IN NS 3dns.losangeles
3dns.newyork IN A 192.168.101.2
3dns.losangeles IN A 192.168.102.2
```

Figure C.2 *Sample code to delegate the domain to each 3-DNS Controller in your network*

3. If your network's master DNS is not a 3-DNS Controller, change (or add) the target domain name to an alias.

   For example, you might find the target domain as an **A** record in your name server's DNS database as follows:

   ```
   www   IN   A   192.168.101.50
   ```

   Edit **db.domain.com** so that it contains the following line:

   ```
   www   IN   CNAME   www.wip
   ```

   In the above line, **www.wip.domain.com** is the domain name controlled by the 3-DNS Controller.

4. You must choose a wide IP key. Select one of the virtual servers in the group, and use its IP address as the wide IP key. In this example, **192.168.101.50** is the wide IP key for **www.wip.domain.com**.

5. Configure the load balanced name on the 3-DNS Controller.

   Locate or create a subdomain database file for **wip.domain.com**. Select one IP address from the set and add an **A** record for the **www.wip** domain. Use the IP

address as the wide IP key. In the new **A** record, specify a low TTL value. (You can override the database's global TTL value for an individual name.)

Figure C.3 is an example of an entire zone file. The next to last line is the **A** record:

```
wip.domain.com.   IN   SOA   3dns.newyork.domain.com.
postmaster.domain.com. (
                       1998062914 ; Serial as YYYYMMDDXX
                       3600 ; Refresh
                       900 ; Retry
                       3600000 ; Expire
                     2 ) ; Minimum (default ttl for entire file)
; Domain DNS servers
wip.domain.com.   IN   NS   3dns.newyork.domain.com.
                  IN   NS   3dns.losangeles.domain.com.
; Glue records
3dns.newyork.domain.com.    IN A 192.168.101.2
3dns.losangeles.domain.com. IN A 192.168.102.2
; Mail servers
domain.com.  IN   MX  10 mx.newyork.domain.com.
domain.com.  IN   MX  20 mx.losangeles.domain.com.
; Regular Host
otherbox    IN    A    192.168.101.20
; domain name      TTL         Wide IP key
www  1   IN   A   192.168.101.50
ftp      IN   A   192.168.101.60
```

Figure C.3 *Sample zone file for* ***wip.domain.com***

The following example is provided for reference only. If you need help establishing reverse domains (address-to-name mappings), refer to the O'Reilly ***DNS***

and BIND book mentioned earlier in this chapter. The following sample screens show the reverse domain mapping files on the New York 3-DNS Controller:

```
101.168.192.in-addr.arpa. IN SOA 3dns.newyork.domain.com.
postmaster.domain.com. (
                          1998062914 ; Serial as YYYYMMDDXX
                          3600 ; Refresh
                          900 ; Retry
                          3600000 ; Expire
                          14000 ) ; Minimum

101.168.192.in-addr.arpa. IN NS 3dns.newyork.domain.com.
                           IN NS 3dns.losangeles.domain.com.

20          IN PTR otherbox.wip.domain.com.
50          IN PTR www.wip.domain.com.
60          IN PTR ftp.wip.domain.com.
```

Figure C.4 *Excerpt from* ***db.192.168.10***

◆ Note

Because a virtual server is listed in each data center for a wide IP definition, you need to define an entry to mapping for each class C network that is included in the wide IP definition.

```
102.168.192.in-addr.arpa. IN SOA 3dns.newyork.domain.com.
postmaster.domain.com. (
                          1998062914 ; Serial as YYYYMMDDXX
                          3600 ; Refresh
                          900 ; Retry
                          3600000 ; Expire
                          14000 ) ; Minimum

102.168.192.in-addr.arpa. IN NS 3dns.newyork.domain.com.
                           IN NS 3dns.losangeles.domain.com.


50   IN PTR www.wip.domain.com.
60   IN PTR ftp.wip.domain.com.
```

Figure C.5 *Excerpt from* ***db.192.168.102***

```
103.168.192.in-addr.arpa. IN SOA 3dns.newyork.domain.com.
postmaster.domain.com. (
                          1998062914 ; Serial as YYYYMMDDXX
                          3600 ; Refresh
                          900 ; Retry
                          3600000 ; Expire
                          14000 ) ; Minimum

103.168.192.in-addr.arpa. IN NS 3dns.newyork.domain.com.
                           IN NS 3dns.losangeles.domain.com.


50          IN PTR www.wip.domain.com.
60          IN PTR ftp.wip.domain.com.
```

Figure C.6 *Excerpt from* ***db.192.168.103***

Instead of a typical one-to-one relationship, where one address maps to one name, the following addresses all map to **www.wip**:

```
192.168.101.50
192.168.102.50
192.168.103.50
```

6. Add the **www.wip.domain.com** domain as a wide IP to your **wideip.conf** file.
7. Define which load balancing mode you want to use for the wide IP. For information on load balancing, see Chapters 5 and 6.
8. List which virtual servers are to be available for load balancing this wide IP.

Figure C.7 provides an example of a **wideip** statement to add to the **wideip.conf** file:

```
wideip {
   address           192.168.101.50
   service           "http"
   name              "www.wip.domain.com"
   ttl               60      // increase the domain default ttl
   pool {
      name           "Pool_1"
      ratio          2      // applies to pool_lbmode == ratio
      preferred      leastconn
      alternate      ratio
      address       192.168.101.50   ratio 2
      address       192.168.102.50   ratio 1
      address       192.168.103.50   ratio 1
     }
   pool {
      name           "Pool_2"
      type           VSb
      ratio          1
      preferred      rr
      address       192.168.102.60   ratio 2
      address       192.168.103.60   ratio 1
   }
}
```

Figure C.7 *Sample wideip statement*

The wide IP is now in place and configured.

To add additional wide IPs by manually editing the configuration file

The following example describes how to add a wide IP named **ftp.wip.domain.com**:

1. Select a set of geographically distributed virtual servers.
2. Select the IP address of one of the virtual servers in the set to be the wide IP key (see *Understanding the wide IP key,* on page 2-27, for details).

3. Define the wide IP name and key within BIND by adding the following resource record to **db.wip.domain.com**:

   ```
   ftp.wip   IN   A   192.168.102.60
   ```

4. Define the virtual server list and the wide IP key within the 3-DNS Controller by adding it to the **wideip.conf** file as follows:

```
wideip {
   address           192.168.102.60
   service           "ftp"
   name              "ftp.wip.domain.com"
   pool {
      name           "main_pool"
      type           VSb
      preferred      leastconn
      alternate      ratio
      address       192.168.101.60   ratio 2
      address       192.168.102.60   ratio 4
      address       192.168.103.60
   }
}
```

Figure C.8 *Sample* ***wideip*** *statement*

5. Restart the 3-DNS Controller by typing the following:

   ```
   ndc restart
   ```

Understanding zone minimums

The zone file contains a Minimum field in the SOA (Start of Authority) section of the file. The Minimum value is the TTL (time to live) for all resource records (RR) in the zone file. However, you can override the zone minimum for a given RR.

For example, if you do not want a DNS to cache the previously issued answer for a domain name, you can specify a very low value for the Minimum field.

◆ Note

*For wide IP domain names, specify the TTL in the **wideip** statement. See The wide IP statement, on page A-37.*

In the following zone file excerpt, the specified Minimum value is 30 seconds for every entry. The exception is the domain name **www.wip**, which is overridden and not saved in any DNS cache. The result is that a new query is made each time there is a name resolution request for **www.wip**. This allows the 3-DNS Controller to respond with the most intelligent answer for each request.

```
wip.domain.com. IN SOA 3dns.newyork.domain.com.
postmaster.domain.com.(
                      1998062914 ; Serial as YYYYMMDDXX
                      3600 ; Refresh
                      900 ; Retry
                      3600000 ; Expire
                    30 ) ; Minimum (default ttl for entire file)
www.wip               0      IN          A      192.168.101.60
```

Figure C.9 *Zone minimums*

Replacing your DNS servers with 3-DNS Controllers as master DNS servers for your domain

If you are replacing your master DNS with 3-DNS Controllers, the method of importing your old zone files is dependent on which option you chose during your planning stage in converting existing BIND files.

If you skipped NameSurfer configuration in the First-Time Boot utility

The **config_namesurfer** script will convert your old zone files and configuration file. The BIND files must be in place before you configure the NameSurfer application.

To transfer and convert existing BIND files

1. Manually transfer the zone files listed as master zones in **/etc/named.conf** on the existing BIND server to **/var/namedb/*** on the principal 3-DNS Controller. To transfer files using crypto 3-DNS Controllers, you can use **scp,** as well as **rcp** or **ftp** in passive mode. To transfer files using non-crypto 3-DNS Controllers, you can use only **rcp** or **ftp** in passive mode.

2. On the 3-DNS Controller, run the **config_namesurfer** script using the command below. Respond **Yes** when asked if you want NameSurfer to be the master.

   ```
   config_namesurfer
   ```

   After the script completes and the NameSurfer application starts, the application automatically converts the transferred zone files from BIND format to NameSurfer format and converts the **/etc/named.conf** file to use the NameSurfer zone files.

The next step is dependent on whether you are using the advanced synchronization features of the 3-DNS Controller and intend to use multiple master DNS servers.

If you intend to use multiple master DNS servers without synchronization

Repeat steps 1 and 2 listed in the previous section, *To transfer and convert existing BIND files*, on each subsequent 3-DNS Controller.

Note

Any changes you make on one 3-DNS Controller must be made on all the others.

If you intend to use multiple master DNS servers with synchronization

1. Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **Yes** when asked if you want NameSurfer to run as a master.
2. On all master 3-DNS Controllers, manually edit the **/var/3dns/etc/sync_list.mandatory** file and uncomment the line:

   ```
   fixed backup /etc/named.conf 3ndc reload
   ```

3. Return to the principal 3-DNS Controller and add the remaining 3-DNS Controllers to the sync group.

Once you finish setting up all of the 3-DNS Controllers and adding each of them to the sync group, the NameSurfer application automatically broadcasts the master zone files to each 3-DNS Controller in the group. All changes that are made with NameSurfer are automatically broadcast to all other 3-DNS Controllers in your sync group. In addition, zone file changes may now be made on any of the 3-DNS Controllers.

If you intend to use one primary 3-DNS Controller with one or more secondary 3-DNS Controllers without synchronization

Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **No** when asked if you want NameSurfer to run as a master.

Note

In this configuration you must perform all your zone edits only on the primary DNS server, which is now the 3-DNS Controller.

If you intend to use one primary 3-DNS Controller with one or more secondary 3-DNS Controllers with synchronization

Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **Yes** when asked if you want NameSurfer to run as a master.

◆ Note

In this configuration you can perform zone edits on your secondary DNS server, and these changes will be broadcast to the primary DNS server. Normal BIND zone transfers will occur. If your primary DNS server becomes unavailable, your zone edits will not be broadcast.

Running 3-DNS Controllers as DNS masters only for wide IP sub-domains

3-DNS Controllers must be the authority for the zone files associated with wide IP definitions. When you set up a configuration where the 3-DNS Controllers are DNS master only for those sub-domains, you must make a few changes to the zone files on the master DNS for your domain. Refer to the section *Relating BIND information to 3-DNS Controller wide IP definitions*, on page C-10.

The next step is dependent on whether you are using the advanced synchronization features of the 3-DNS Controller and intend to use multiple master DNS servers.

If you intend to use multiple master DNS servers without synchronization

Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **Yes** when asked if you want NameSurfer to run as a master.

◆ Note

Any changes you make on one 3-DNS Controller must be made on all the others.

If you intend to use multiple master DNS servers with synchronization

1. Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **Yes** when asked if you want NameSurfer to run as a master.
2. On all master 3-DNS Controllers manually edit the **/var/3dns/etc/sync_list.mandatory** file and uncomment the line:

   ```
   fixed backup /etc/named.conf 3ndc reload
   ```

3. Return to the principal 3-DNS Controller and add the remaining 3-DNS Controllers to the sync group.

Once you finish setting up all of the 3-DNS Controllers and adding each of them to the sync group, the NameSurfer application automatically broadcasts the master zone files to each 3-DNS Controller in the group.

If you intend to use one primary 3-DNS Controller with one or more secondary 3-DNS Controllers without synchronization

Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **No** when asked if you want NameSurfer to run as a master.

◆ Note

In this configuration you must perform all your zone edits only on the primary DNS server, which is now the 3-DNS Controller.

If you intend to use one primary 3-DNS Controller with one or more secondary 3-DNS Controllers with synchronization

Run the First-Time Boot utility on the remaining 3-DNS Controllers in the network. Make sure you choose **Yes** when asked if you want NameSurfer to run as a master.

◆ Note

In this configuration you can perform zone edits on your secondary DNS server, and these changes will be broadcast to the primary DNS server. Normal BIND zone transfers will occur. If your primary DNS server becomes unavailable, your zone edits will not be broadcast.

F5 Networks

3·DNS®

Distributed Traffic Controller

D

DNS Resource Records

Overview

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 DNS.

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

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

The 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 will be stored in the database. If this field is left blank, the default time to live value is specified in the Start Of Authority 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 case insensitive.

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.

Common types

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 D.1 *Standard resource records*

A (Address)

The Address record, or **A** name 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** name record for each address of the machine.

The following is an example of an **A** name record:

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

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.

The following is an example of a CNAME resource record:

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

MX (Mail Exchange)

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

The following is 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.
```

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.

The following is an example of an NS resource record:

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

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 following example of a PTR record is used in setting up reverse pointers for the special IN-ADDR.ARPA domain:

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

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    ucbvax.Berkeley.Edu.
    kjd.ucbvax.Berkeley.Edu. (
                               1995122103    ; Serial
                               10800         ; Refresh
                               1800          ; Retry
                               3600000       ; Expire
                               259200 )      ; Minimum
```

The record-specific fields are defined as follows:

- **Person in charge**
  The email address for the person responsible for the name server, with "@" changed to a "."
- **Serial number**
  The version number of this data file; it must be a positive integer. This number must be increased 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 see 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.

Other types

The following is a list of less common resource record types. For more information see RFCs 1035, 1183, and 1664.

Type	Description
AAAA	IPv6 address
AFSDB	AFS database location
GPOS	Geographical position
HINFO	Host information
ISDN	Integrated services digital network address
KEY	Public key
KX	Key exchanger
LOC	Location information
MB	Mailbox domain name
MINFO	Mailbox or mail list information
NULL	A null RR

Table D.2 *Other types of resource records*

Type	Description
NSAP	Network service access point address
NSAP-PTR	(Obsolete)
NXT	Next domain
PX	Pointer to X.400/RFC822 information
RP	Responsible person
RT	Route through
SIG	Cryptographic signature
SRV	Server selection
TXT	Text strings
WKS	Well-known service description
X25	X25

Table D.2 *Other types of resource records*

F5 Networks

3·DNS®

Distributed Traffic Controller

Glossary

3-DNS Maintenance menu

A command line utility that you can use to manually configure the 3-DNS Controller.

3-DNS web server

A standard web server that runs on the 3-DNS Controller and hosts the Configuration utility, and also provides access to useful downloads.

A record

The answer that a 3-DNS Controller returns to a local DNS server in response to a name resolution request. The **A** record contains a variety of information, including one or more IP addresses that resolve to the requested domain name.

active unit

In a redundant system, the controller that currently load balances connections. If the active unit in the redundant system fails, the standby unit assumes control and begins to load balance connections.

alternate method

Specifies the load balancing mode to use if the primary method fails.

big3d agent

A monitoring agent that collects metrics information about server performance and network paths between a data center and a specific local DNS server. The 3-DNS Controller uses the information collected by the **big3d agent** for dynamic load balancing.

BIND (Berkley Internet Name Domain)

The most common implementation of DNS, which provides a system for matching domain names to IP addresses.

chain

A series of filtering criteria used to restrict access to an IP address. The order of the criteria in the series determines how the filter is applied, from the general criteria first, to the more detailed criteria at the end of the chain.

Configuration utility

The browser-based application that you use to configure the 3-DNS Controller.

completion rate

The percentage of packets that a server successfully processes during a given conversation.

data center

A physical location that houses one or more 3-DNS Controllers, BIG-IP Controllers, or host machines.

discovery factory

A tool managed by the **big3d** agent that checks for alternate ports to ping when trying to collect path data for a local DNS.

dynamic load balancing modes

Dynamic load balancing modes base connection distribution on live data, such as current server performance and current connection load.

dynamic ratio

An option for the Quality of Service load balancing mode. The dynamic ratio feature uses QOS scores as ratios for virtual servers and distributes connections according to the ratio weight for each virtual server.

F-Secure SSH

An encryption utility that allows secure shell connections to a remote system.

fail-over

The process in which a standby unit in a redundant system takes over due to a software or hardware failure detected on the active unit.

fail-over cable

The cable that directly connects the two controller units in a redundant system.

FDDI (Fiber Distributed Data Interface)

A multi-mode protocol for transmitting data on optical-fiber cables up to 100 Mbps.

First-Time Boot utility

A utility that walks you through the initial system configuration process. The First-Time Boot utility runs automatically when you turn on a controller for the first time.

fallback method

Specifies the last load balancing mode that the 3-DNS Controller tries to use if both the primary and the alternate methods fail.

hardware-based fail-over

A redundant system in which the two units are connected directly by a cable.

hops

One point-to-point transmission in a network path between a host and a client server. A network path that included a stop at a network router would have two hops: the first from the client to the router, and the second from the router to the host server.

hops factory

A type of factory run by the **big3d** agent that collects hops data for network paths.

host

A network server which manages one or more virtual servers that the 3-DNS Controller uses for load balancing.

ICMP (Internet Control Message Protocol)

An Internet communications protocol used to determine information about routes to destination addresses, such as virtual servers managed by BIG-IP Controllers and 3-DNS Controllers.

IP filter

A filter that specifically allows or denies traffic passing through an interface on the 3-DNS Controller, based on either the source IP address, the destination IP address, or both.

iQuery

A UDP-based protocol used to exchange information between **big3d** agents and 3-DNS Controllers. The iQuery protocol is officially registered for port 4353.

LDNS probe state

The status of an local DNS server with respect to metrics collection.

LDNS Round Robin

A standard DNS feature that allows a DNS server, or a 3-DNS Controller, to return multiple IP address in the **A** record. The local DNS, or even the browser, can cache the IP addresses and use them for future name resolutions.

local DNS (LDNS)

A local DNS server typically found at a client's Internet service provider. The 3-DNS Controller calculates path information for the path between the local DNS and the virtual servers that the 3-DNS Controller is load balancing, and also uses the local DNS IP address for topology-based load balancing and access control.

master DNS

A DNS server that is considered authoritative for one or more DNS zones. See also *zones*.

metrics

Performance data, including server performance and network path integrity, collected by **big3d** agents and used by the 3-DNS Controller for dynamic load balancing.

named

The name server daemon, which manages domain name server software.

NameSurfer

The third-party application that automatically manages DNS zone files on 3-DNS Controllers.

network-based redundant system

A redundant system in which two units communicate over a network connection, rather than a hard-wired connection.

path

The network route between two specific IP addresses. For dynamic load balancing, the 3-DNS Controller uses information about the path between a client LDNS and a specific virtual server that it is load balancing.

packet rate

The number of packets per second going in or coming out of a given server.

pool

A group of virtual servers to which the 3-DNS Controller distributes connections when load balancing a specific domain.

port

A number that is associated with a specific service supported by a host.

preferred method

Specifies the primary load balancing mode used to load balance a wide IP.

principal 3-DNS Controller

A 3-DNS Controller that initiates metrics collection by the **big3d** agents. Note that a sync group can have only one principal.

probing factory

A tool managed by the **big3d** agent that queries virtual servers to determine whether they are **up** or **down**, and also to determine path metrics such as round trip time and hops.

production rule

A tool that can change system behavior under specific operating conditions. For example, a production rule can switch load balancing modes or can reroute network traffic to a specific set of servers based on triggers such as time of day or current network traffic load.

QOS coefficient

The ratio weight for a specific factor used in a QOS equation, such as hops, round trip time, packet rate, completion rate, or topology score. Each factor has a default coefficient of 1, but you can assign user-defined coefficients to put more weight on a specific factor, and less on other factors.

QOS equation

A calculation based on various path statistics used for dynamic load balancing including hops, round trip time, packet rate, completion rate, and topology score.

QOS score

The result of a QOS calculation. When using dynamic load balancing modes, the 3DSN Controller uses the QOS score to determine which virtual server has the best performance and should receive new connections.

ratio

A parameter that assigns a weight to a virtual server for load balancing purposes.

Ratio mode

The Ratio load balancing mode distributes connections across an array of virtual servers in proportion to the ratio weights assigned to each individual virtual server.

receiver 3-DNS Controller

A 3-DNS Controller that receives broadcasted metrics data from **big3d** agents, but does not initiate metrics collection (see *principal 3-DNS Controller*).

redundant system

A pair of controllers that are configured for fail-over. In a redundant system, there are two controller units, one running as the active unit and one running as the standby unit. If the active unit fails, the standby unit takes over and manages connection requests.

remote administrative IP address

An IP address from which a controller allows shell connections, such as Telnet or SSH.

rsetup

The script that configures the remote login tools, **rsh** and **rcp**, on a controller.

slave DNS server

A DNS server that updates its records by querying a master DNS server.

shared IP alias

The IP address that represents a redundant system.

SNMP (Simple Network Management Protocol)

A standard Internet standard protocol developed to manage nodes on an IP network.

SNMP agent

The agent that controls SNMP communications on a server.

SNMP factory

A type of factory run by **big3d** agents that uses the SNMP protocol to collect metrics data for host servers.

SNMP MIB (Management Information Base)

A text file in standard SNMP format that defines the individual objects you can manage with common SNMP tools such as HP OpenView.

sod (switch over daemon)

A daemon that controls the fail-over process in a redundant system.

standby unit

A controller in a redundant system that is always prepared to become the active unit if the active unit fails.

static load balancing mode

A static load balancing mode bases connection distribution on a pre-defined list of criteria; it does not take current server performance or current connection load into account.

sync group

A group of 3-DNS Controllers that share configuration information. Each sync group has one principal 3-DNS Controller, and can also have one or more receiver 3-DNS Controllers.

time tolerance value

The number of seconds that one 3-DNS Controller's clock is allowed to differ in comparison to another 3-DNS Controller's clock, without the two clocks being considered out of sync.

topology record

A record that specifies a score for a specific pair of a local DNS IP address and a virtual server IP address or range of IP addresses. Topology records are used for topology-based access control, and topology load balancing modes.

topology score

A rank for a specific pair of a local DNS server and a virtual server IP address, or range of IP addresses.

topology-based access control

A method of specifically allowing or preventing clients from connecting to particular virtual servers based on the IP address of the requesting client's local DNS and the IP address of the virtual server.

traceroute

The utility that the hops factory uses to calculate total number of network hops between an LDNS and a specific data center.

TTL (time-to-live)

The number of seconds for which a specific DNS record or metric is considered to be valid. When a TTL expires, the server usually must refresh the information before using it.

virtual server

A specific combination of a virtual address and virtual port, associated with a content site that is managed by a BIG-IP Controller or other type of host server.

watchdog timer card

A hardware device that monitors the 3-DNS Controller for hardware failure.

well known services (WKS)

A type of resource record that describes the services usually provided by a particular protocol on a particular port.

wide IP

A mapping of a fully-qualified domain name to a set of virtual servers that host the domain content, such as a web site or an e-commerce site.

wide IP key

An IP address associated with a wide IP. The wide IP key is the same address that standard DNS associates with the domain name the wide IP load balances.

zone

A subset of DNS records for one or more domains.

zone file

A set of DNS records that contain domains with one or many domain names, designated mail servers, a list of other name servers that can answer resolution requests, and a set of zone attributes called SOA (Start Of Authority).

F5 Networks

3·DNS®

Distributed Traffic Controller

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z