HP DNS Configuration and Management Manual

Abstract

This manual describes how to implement Domain Name System (DNS) 9.2.3 on the NonStop Server in the HP NonStop Open System Services (OSS) environment. This manual is intended for network administrators responsible for managing DNS.

Product Version

DNS 9.2.3

Supported Release Version Updates (RVUs)

This publication supports G06.25 and all subsequent RVUs until otherwise indicated in a new edition.

Part Number	Published
529432-001	December 2004

Document History

Part Number	Product Version	Published
529432-001	DNS 9.2.3	December 2004

HP DNS Configuration and Management Manual

Glossary | Index | Examples | Figures | Tables

3. Configuring and Managing DNS 9.2.3 (continued)

4. Troubleshooting

Glossary

Index

Examples

Figures

Tables

What's New in This Manual

Manual Information

Abstract

This manual describes how to implement Domain Name System (DNS) 9.2.3 on the NonStop Server in the HP NonStop Open System Services (OSS) environment. This manual is intended for network administrators responsible for managing DNS.

Product Version

DNS 9.2.3

Supported Release Version Updates (RVUs)

This publication supports G06.25 and all subsequent RVUs until otherwise indicated in a new edition.

Part Number	Published
529432-001	December 2004

Document History

Part Number	Product Version	Published
529432-001	DNS 9.2.3	December 2004

New and Changed Information

This is a new manual.

About This Manual

This manual describes how to implement Domain Name System (DNS) 9.2.3 on NonStop Server in the HP NonStop Open System Services (OSS) environment. For general information about DNS 9.2.3, see the *BIND 9 Administrator Reference Manual* in the NonStop Technical Library (NTL).

This section contains the following information:

Manual Organization

Table i. Manual Organization

Section	Description
Section 1, Quick Start	Provides examples of saving your existing named.conf file, starting DNS 9.2.3, and stopping DNS 9.2.3.
Section 2, Introduction to DNS 9.2.3	Provides a brief introduction to DNS 9.2.3, including restrictions, assumptions, and tools.
Section 3, Configuring and Managing DNS 9.2.3	Provides configuration and management information about DNS 9.2.3, including configuring the named process as a persistent process, how to use the tools, and how to specify a different resolv.conf file.
Section 4, Troubleshooting	Provides guidelines for troubleshooting problems with DNS 9.2.3

Supported RFCs

The following Requests for Comments (RFCs) are supported by DNS 9.2.3 Release 1:

- **RFC 1886**: DNS Extensions to Support IPv6
    - The changes include a new resource record (RR) type to store an IPv6 address, a new domain to support lookups based on an IPv6 address, and

    updated definitions of existing query types that return Internet addresses as part of additional section processing.
    - The extensions are compatible with existing applications and, in particular, DNS implementations themselves.

- **RFC 2874**: DNS Extensions to Support IPv6 Address Aggregation and Renumbering
    - Changes are made to DNS to support aggregated and renumbered IPv6 addressing.
    - A new RR type, A6, maps a domain name to an IPv6 address, with a provision for indirection for leading prefix bits.
    - A new domain, IP6.ARPA, supports lookups based on IPv6 addresses.
    - The new scheme for reverse lookups relies on a new type of DNS label called the bit-string label [BITLBL], which represents an arbitrary string of bits that is treated as a hierarchical sequence of one-bit domain labels.
    - DNS address-space delegation is implemented by a new analogue to the CNAME record, called the DNAME RR, which provides alternate naming to an entire subtree of the domain-name space, rather than to a single node.
- **RFC 2845**: Secret Key Transaction Authentication for DNS (TSIG)
    - Allows for transaction-level authentication using shared secrets and one-way hashing.
    - It can be used to authenticate dynamic updates as coming from an approved client, or to authenticate responses as coming from an approved recursive Name Server.

Statement for Ported Software and Related Documentation

TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Assumptions About the Reader

- You are familiar with, or have knowledge of, previous versions of DNS
- You are a network administrator who works with HP NonStop systems on a regular basis

Related Manuals

- *Open System Services User's Guide*
- *Open System Services Installation Guide*
- *Open System Services Management and Operations Guide*
- man pages for `nsupdate`, `rndc`, and `named`

About the *BIND 9 Administrator Reference Manual*

The *BIND 9 Administrator Reference Manual* provides basic information about the concepts and management of the Internet Software Consortium (ISC) BIND version 9 software package for system administrators, and includes the following topics:

- Introduction to basic DNS and BIND concepts
- More advanced concepts that may be needed for implementing certain options
- A reference section to aid in the ongoing maintenance of the software
- Security considerations and troubleshooting help

In some instances, the BIND manual contains information that is not supported as of the G06.25 RVU (DNS 9.2.3 Release 1) such as:

- Domain Name Security Extensions (DNSSEC)
- Lightweight Resolver

As these features are not supported for Release 1 of DNS 9.2.3, they should be ignored in the BIND manual.

Notation Conventions

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

This requirement is described under Backup DAM Volumes and Physical Disk Drives on page 3-2.

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS. Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
MAXATTACH
```

lowercase italic letters. Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

file-name

computer type. `Computer type` letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
myfile.c
```

italic computer type. *`Italic computer type`* letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

`pathname`

[ ] Brackets. Brackets enclose optional syntax items. For example:

```
TERM [\system-name.]$terminal-name

INT[ERRUPTS]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num  ]
   [ -num ]
   [ text ]

K [ X | D ] address
```

{ } Braces. A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name  }
```

```
ALLOWSU { ON | OFF }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

… Ellipsis. An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]…
```

```
[ - ] {0|1|2|3|4|5|6|7|8|9}…
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char…"
```

Punctuation. Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing. Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing. If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE

   [ , attribute-spec ]…
```

!i and !o. In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT (  segment-id                          !i
                         , error       ) ;                     !o
```

!i,o. In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;                            !i,o
```

!i:i. In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ (  filename1:length                 !i:i
                            , filename2:length ) ;             !i:i
```

!o:i. In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ (  filenum                              !i
                        , [ filename:maxlen ] ) ;              !o:i
```

Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text. Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE

?123

CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic text. Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

lowercase italic letters. Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

p-register

process-name

[ ] Brackets. Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
proc-name trapped [ in SQL | in SQL file system ]
```

{ } Braces. A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
obj-type obj-name state changed to state, caused by
{ Object | Operator | Service }
```

```
process-name State changed from old-objstate to objstate
{ Operator Request. }
{ Unknown.          }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% Percent Sign. A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
```

```
%B101111
```

```
%H2F
```

```
P=%p-register E=%e-register
```

Change Bar Notation

Change bars are used to indicate substantive differences between this manual and its preceding version. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

> The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).
>
> The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

Abbreviations

BIND	Berkeley Internet Name Domain
DNS	Domain Name System
ISC	Internet Systems Consortium
IXFR	Incremental Zone Transfer
OSS	Open System Services

1 Quick Start

This section provides examples of preparing to start Domain Name System (DNS) 9.2.3, starting DNS 9.2.3, and stopping DNS 9.2.3. This section does not provide background information on DNS or NonStop TCP/IPv6; refer to the following manuals (located in the NonStop Technical Library (NTL)) to gain a more in-depth understanding of DNS and NonStop TCP/IPv6, respectively:

- *BIND 9 Administrator Reference Manual*
- *TCP/IPv6 Configuration and Management Manual*

This section provides the following procedures:

- I. Prepare to Start DNS 9.2.3 on page 1-1
- II. Start DNS 9.2.3 on page 1-2
- III. Stop DNS 9.2.3 on page 1-2

I. Prepare to Start DNS 9.2.3

1. Check Assumptions Before Starting DNS 9.2.3

Before starting DNS 9.2.3, check that the following assumptions are met:

- The Open System Services (OSS) subsystem is available. Refer to the *Open System Services Installation Guide* in NTL for installation procedures, and the *Open System Services Management and Operations Guide* in NTL for operating procedures.
- DNS 9.2.3 is installed. To check that DNS 9.2.3 is installed, perform the following steps:

  a. Change the directory to /etc/dns923/ in the OSS shell as follows:

```
>cd /etc/dns923/
```

  b. List the files in the directory by entering the following command:

```
/etc/dns923>ls -al
```

  You should see a display that shows a list of DNS files along with their permissions. These files include:

  - named
  - named-bootconf.sh
  - named.conf
  - nsupdate

        - rndc
        - rndc.conf
        - db.myzone.com.in

- NonStop TCP/IP is running. If you plan to use IPv6 communications, verify that NonStop TCP/IPv6 is running in DUAL or INET6 mode; refer to the *TCP/IPv6 Configuration and Management Manual* in NTL for more information.
- You are the super ID (equivalent to a UNIX root user)
- You have the following software components to run a Name Server:
    - DNS 9.2.3 name daemon
    - Valid configuration file (named.conf)
    - Zone files

II. Start DNS 9.2.3

Start DNS 9.2.3 with default options, which is the simplest way of starting the `named` process. The zone files are automatically loaded from the location specified in the named.conf file, which is looked up by default in /etc. Since the named.conf file is located in the /etc/dns923 directory, you must use the **-c** option in conjunction with the `named` command.

1. Enter the following command at the OSS prompt to start `named`:

```
/etc/dns923>named -c /etc/dns923/named.conf
```

2. Enter the following command to check that the `named` process is running:

```
/etc/dns923>ps | more
```

III. Stop DNS 9.2.3

The DNS 9.2.3 server (`named` process) can be stopped in two ways:

- Using the tool `rndc` (see rndc (remote name daemon control) on page 2-3)
- Using the OSS `kill` command

  Signals such as SIGINT and SIGTERM can be passed to the `named` process using the `kill` command, causing `named` to be stopped.

1. Enter the following command to obtain the process ID for the `named` process:

```
/etc/dns923>ps | more
```

2. Enter the following command at the prompt to stop `named`:

```
/etc/dns923>kill -term process-id
```

The above command sends the SIGTERM signal to the `named` process, causing the process to terminate.

Note. If `named` is a persistent process, see Managing Persistence for the named Process on page 3-1; the `kill` command does not stop the `named` process.

2 Introduction to DNS 9.2.3

This section serves as a brief introduction to the Domain Name System (DNS) 9.2.3 process. For a more thorough discussion of DNS 9.2.3, see the *BIND 9 Administrator Reference Manual* in the NonStop Technical Library (NTL).

Overview

BIND (Berkeley Internet Name Domain) is an implementation of the DNS protocols and provides an implementation of the major components of the DNS, including:

- A Domain Name System server *(*named process)
- A Domain Name System resolver library
- Tools for verifying the proper operation of the DNS server

DNS 9.2.3 is a port of BIND 9.2.3, and no architectural changes were made to the product during the porting.

The system that provides the name-resolution feature (to resolve names into IP addresses) is called the Name Server. The core process of DNS is a daemon called named. Depending on the role assigned, the Name Servers can be Primary (Master), Secondary (Slave), or Caching-only. The Secondary server takes over when the Primary is down and is updated automatically. A Caching-only server provides only the caching information to the clients.

Each domain, or the sub-domain, has information (in zone files or data files) about its hosts and is called Authoritative for these hosts. For hosts about which the domain does not have any information or for which it is Non-Authoritative, the domain passes the query to its parent domain.

The clients (hosts) know about their Name Servers through a file called resolv.conf, which contains addresses of the Name Servers (Primary, Secondary, and Caching-only) along with the clients’ domain name. The configuration file of the Name Server is named.conf, which contains server parameters and references to other data files containing the host information.

Restrictions and Features

Restrictions

- To make DNS a persistent process using the NonStop Kernel Persistence Manager ($ZZKRN), you should have the G06.24 RVU installed on the system.

Features

- DNS Protocol Enhancements

  Incremental Zone Transfer (IXFR), NOTIFY, and dynamic updates are used for efficient zone transfer, reducing size and duration, and supporting the increasing amount of data being exchanged between Name Servers.

- IPv6 Address Resolution

  The A6 Resource Record (RR) is used for address translation between a domain name and an IPv6 address, and can be used to replace the currently used AAAA RR.

- The following RFCs are supported:
  - RFC 1886: DNS Extensions to Support IPv6
  - RFC 2874: DNS Extensions to Support IPv6 Address Aggregation and Renumbering
  - RFC 2845: Secret Key Transaction Authentication for DNS (TSIG)

Assumptions and Dependencies

- Applications running in either OSS or Guardian can access the Name Server but the `named` process must run in OSS.
- For the Name Server to listen to IPv6 addresses, NonStop TCP/IPv6 must be running in DUAL or INET6 mode (refer to the *TCP/IPv6 Configuration and Management Manual* in NTL for more information).
- $ZZKRN is used to make DNS 9.2.3 run as a persistent process.

Tools

The following tools are supported as part of the DNS 9.2.3 product (for more information about a tool, refer to the *BIND 9 Administrator Reference Manual* in NTL).

named-bootconf

The `named-bootconf` tool is a shell script that converts a DNS 4.8.3 configuration file to a DNS 9.2.3 configuration file.

The syntax for this tool is:

```
/etc/dns923> named-bootconf.sh < DNS 4.8.3 configuration file
(input) > DNS 9.2.3 configuration file (output)
```

Refer to Using the named-bootconf Utility to Migrate the Configuration File on page 3-2 for an example of using the `named-bootconf` tool.

Note. "<" and ">" are redirection operators. Refer to Using the nsupdate Tool on page 3-3.

nsupdate

The `nsupdate` tool updates DNS 9.2.3 domain Name Servers that support dynamic updates; `nsupdate` uses the DNS resolver library to pass messages to a Name Server, requesting the insertion or deletion of DNS RRs.

The syntax for this tool is:

```
/etc/dns923> nsupdate [-k keydir:keyname] [-d] [-v] [filename]
```

where:

-k option sign updates with TSIG

-d sets the debug mode

-v tells `nsupdate` to use TCP/IP to communicate with the server

Refer to Using the nsupdate Tool on page 3-3 for an example of using the `nsupdate` tool.

rndc (remote name daemon control)

The `rndc` tool allows you to control the operation of the Name Server, either locally or remotely. The `rndc` tool uses the /etc/rndc.conf file for its configuration options, which can be overridden by using command-line options.

The syntax for this tool is:

```
/etc/dns923> rndc [-c config] [-s server] [-p port] [-y key]
[command]
```

Some of the supported commands are:

- **reload**: Reloads configuration file and zones.
- **stats**: Writes server statistics to the statistics file.
- **stop**: Stops the server, making sure any recent changes made through dynamic update or IXFR are first saved to the master files of the updated zones.

- **halt**: Stops the server immediately. Recent changes made through dynamic update or IXFR are not saved to the master files, but are rolled forward from the journal files when the server is restarted.
- **status**: Displays the status of the server.

Refer to Using the rndc Tool on page 3-3 for an example of using the `rndc` tool. Refer to the *BIND 9 Administrator Reference Manual* in NTL for additional supported commands and additional information about `rndc`.

3

Configuring and Managing DNS 9.2.3

Managing Persistence for the named Process

The NonStop Kernel Persistence Manager ($ZZKRN) provides support for making HP NonStop Open System Services (OSS) processes persistent, starting with the G06.24 RVU. This feature of $ZZKRN can be used in making DNS 9.2.3 persistent.

Configuring the named Process as a Persistent Process

Use the Subsystem Control Facility (SCF) to configure the Name Server (named process) to $ZZKRN. Perform this procedure in the Guardian environment.

1. Add named to the system configuration database:

```
->add process $zzkrn.#named, name $osh, autorestart 10, &
cpu firstof (1,0), startmode manual, userid super.super, &
program $system.system.osh, asocproc $dns, &
hometerm $ztn0.#ptypt2q, startupmsg &
“-ls -name /g/dns -p /usr/bin/named -f”
```

   For information about the attributes shown in this command example, see the *SCF Reference Manual for the Kernel Subsystem* in NTL.

2. Start the $ZZKRN.#named process:

```
->start process $zzkrn.#named
```

Stopping the named Process as a Persistent Process

1. To stop the `named` process if it is configured as a persistent process (`autorestart` > 0), issue the SCF ABORT command to the NonStop Kernel subsystem as shown:

```
->abort $zzkrn.#named
```

For more information about managing persistence, see the *SCF Reference Manual for the Kernel Subsystem* in NTL.

Specifying a TCP/IP Process By Using a Runtime Option

By default, the `named` process binds to the TCP/IP process $ztc0. To make the `named` process use the TCP/IP process that is specified at runtime, a runtime option (**-T**) is provided (the **-T** option takes the name of the TCP/IP process that is to be used by `named`).

No checking is done by `named` for the validity of the process name. If you specify an invalid TCP/IP process name, the `named` process will not start.

If you do not specify the **-T** option, `named` gets bound to $ztc0.

To run `named` on the TCP/IP process $ztcp2, issue the following command:

```
/etc/dns923> named -T \$ztcp2
```

Using the named-bootconf Utility to Migrate the Configuration File

The DNS configuration-file syntax has changed significantly from the Version 4.x to the Version 9.x. The DNS 4.x configuration file can be converted to a DNS 9.x configuration file by running the script named-bootconf.sh. This shell script takes the old configuration file as input and produces the new syntax as output, which can then be redirected to a file.

In the following example, the name of the old configuration file (DNS 4.x format) is named.boot, and the name of the new generated file (DNS 9.x format) is named.conf.

Example 3-1. Migrating the Configuration File

```
/etc/dns923> named-bootconf.sh < named.boot > named.conf
```

In the example above, “<“ and “>” are redirection operators.

Note. Since the manner of specifying pathnames is different in Guardian and OSS, you must ensure that the new configuration file (after conversion) contains the appropriate path of the zone files.

Using the nsupdate Tool

The `nsupdate` tool can be used to dynamically update the DNS 9.2.3 Name Server. The `nsupdate` tool uses the DNS resolver library to pass messages to a DNS server that request the insertion or deletion of DNS resource records (RR). `nsupdate` can take inputs from a file or the standard input.

The following example illustrates the interactive use of `nsupdate`. In this example, nsupdate connects to a Name Server running on 10.53.0.1, then the existing record `old.example.com` is deleted and a new record `new.example.com` is added.

Example 3-2. Using the `nsupdate` Tool

```
/etc/dns923>nsupdate

>server 10.53.0.1

>update delete old.example.com A

>update add new.example.com 3600 A 10.1.1.1

>

>quit
```

Using the rndc Tool

The `rndc` (remote name daemon control) tool allows you to control the operation of the DNS 9.2.3 Name Server.

In the following example, `rndc` stops the Name Server running on 10.53.0.1.

Example 3-3. `rndc` Command

```
> rndc -s 10.53.0.1 stop
```

The communication between the Name Server and `rndc` is secured by a shared secret key. `rndc` reads a configuration file to determine how to contact the Name Server and decide what algorithm and key to use.

Specifying a Different resolv.conf File

The resolver uses the /etc/resolv.conf file by default.

The new environment variable TCPIP_RESOLVER_NAME must be used for pointing to the resolv.conf file; this variable can be set either through the OSS shell prompt or through a call to the putenv() procedure in the OSS application.

The following pieces of code are examples of resolving the destination host by querying the Name Server pointed at by the /etc/dns/resolv.conf file.

Example 3-4. Specifying a Different resolv.conf File Through an Application

```
putenv ("TCPIP_RESOLVER_NAME=/etc/dns/resolv.conf");

h = gethostbyname(argv[1]);

if(h==NULL) {

   printf("%s: unknown host '%s'\n",argv[0],argv[1]);

      exit(1);

         }
```

Example 3-5. Specifying a Different resolv.conf File Through the OSS Shell

```
/etc/dns923> export

TCPIP_RESOLVER_NAME='/etc/dns/resolv.conf'

/etc/dns923> set
```

Using IXFR

By default, the support for IXFR for Slave Name Servers is active, and can be disabled at configuration time by specifying statements in the zone data file.

The basic syntax of such statements in the configuration file is:

```
Server ip-address {

      provide-ixfr no[yes];

};
```

For example:

```
Server 10.53.0.1 {

      provide-ixfr no;

};
```

The `provide-ixfr` clause determines whether the local server (acting as Master) responds with an incremental zone transfer when the given remote server (a Slave) requests it. If set to `yes` (case sensitive), incremental zone transfer is provided whenever possible. If set to no (case sensitive), all transfers to the remote server are non-incremental.

Performing Dynamic Updates

For dynamic updates to be processed by the Name Server, specify the IP addresses of the systems from where the requests can be received. The IP addresses are specified in the configuration file under the zone entry.

The basic syntax of such an entry in the configuration file is:

```
zone "zone-name " {
      type master;
      file "zone filename";
      allow-update { list of IP-addresses; };
};
```

For example:

```
zone "example.com " {
      type master;
      file "db.example";
      allow-update { 10.53.0.24; };
};
```

This example indicates that the Name Server is a Master for the zone “`example.com`” and the Name Server can update the zone data file “`db.example`” upon receiving the update requests from the system configured with IP address 10.53.0.24.

The `allow-update` clause specifies which hosts are allowed to submit dynamic DNS updates for Master zones. The default is to deny updates from all hosts.

4 Troubleshooting

This section provides guidelines for troubleshooting various problems you may encounter with DNS 9.2.3. This section contains the following information:

- Troubleshooting DNS 9.2.3 on page 4-1
- Logging in DNS 9.2.3 on page 4-3

Troubleshooting DNS 9.2.3

In most cases, the primary cause of the process `named` failing to start is faulty configuration. The best way of troubleshooting this error is to set up logging beforehand; the log files provide hints and information that can be used to determine what went wrong. Refer to Logging in DNS 9.2.3 on page 4-3 for the description of the DNS logging mechanism.

Alternatively, a more interactive and simpler (but not as detailed) way to troubleshoot errors is to start `named` with the `-g` option. This causes the server to be run in the foreground and all logging is forced to `stderr`*.*

The following is a sample named.conf (configuration) file with various syntax errors; line numbers are shown for convenience.

Example 4-1. Sample named.conf File

```
1 /*Sample configuration file */
2
3 options {
4         directory "/user/dns/nameddir";
5         pid-file "named.pid"                  /*semicolon missing*/
6         listen-on { 172.31.45.115; };
7 };
8
9 logging {
10        channel my_syslog {
11                          syslog daemon;
12                          severity info;
13        }                                      /*semicolon missing*/
14
15        channel my_file {
16                        file "log.msgs";
17                        severity dynamic;
18        };
19
20        category statistics { my_ems; my_file; };
21        category queries { my_ems; my_file; };
22 };
23
24 zone "myzone.com" in {
25         type master;
25         file "db.myzone.com";
27 };
28 /*Sample configuration file ends*/
```

When named is run with the -g option using the above configuration, the following is displayed:

```
Aug 30 10:26:14.0lu /etc/named.conf:5: missing ';' before 'pid-file'
Aug 30 10:26:14.0lu /etc/named.conf:13: missing ';' before '}'
Aug 30 10:26:14.0lu loading configuration: failure
Aug 30 10:26:14.0lu exiting (due to fatal error)
```

As shown, the error is displayed along with date, time and line numbers (5 and 13 in this case).

Logging in DNS 9.2.3

There are two main phrases in logging:

- Channels specify where the data is logged
- Categories specify what kind of data (queries, updates, statistics, etc.) has to be logged in specified channels

Standard channels are:

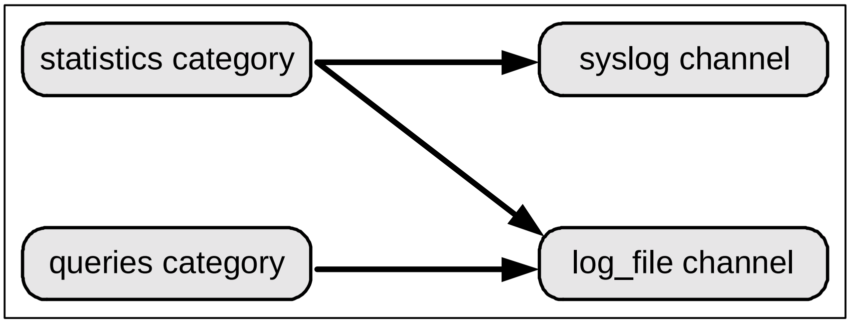
- file
- syslog
- stderr
- null

Standard categories are:

- default
- general
- database
- security
- config
- resolver
- xfer-in
- xfer-out
- notify
- client
- unmatched
- network
- update
- queries
- dispatch
- lame-servers

Each category of data can be sent to one or more channels. In Figure 4-1, queries are logged to a file while statistics data is logged both to a file and to syslog.

Figure 4-1. Logging Categories to Channels



The Name Server can be instructed to enable this feature by specifying the logging statement in the named.conf file.

An example statement in the named.conf file is:

Example 4-2. Logging Statement in named.conf File

```
logging
{
      channel my_ems
      {
           syslog daemon;
           severity info;
      };
      channel my_file
      {
           file "mylog.log";
           severity dynamic;
      };
 category statistics { my_ems; my_file; };
 category queries { my_file; };
};
```

In this example, two channels have been defined:

- `my_ems` instructs the Name Server to send logging data to the Event Management System (EMS)
- `my_file` instructs the Name Server to send logging data to the file `mylog.log`

Similarly, two categories have been defined:

- `statistics` instructs the Name Server to send the statistical data to both EMS and a local file
- `queries` instructs the Name Server to send the logging data on queries to a local file only

Refer to the *BIND 9 Administrator Reference Manual* in NTL for additional information on the logging statement, channels, and categories.

Glossary

Advanced Research Projects Agency (ARPA). An agency of the United States Department of Defense, ARPA underwrote the development of the Internet beginning in 1969. Known as ARPANET, it was designed so that, in case of war and the loss of any group of sites, remaining sites would still be able to communicate along alternate routes. No site would be critical to the operation of the network. Eventually, ARPANET was divided into Milnet, which connected military sites, and a new ARPANET that connected other sites, mainly universities. A new communication protocol was developed, TCP/IP, so that all sites on either of the networks could communicate.

ARPA. See Advanced Research Projects Agency (ARPA).

Berkeley Internet Name Domain (BIND). An implementation of the DNS protocols that provides an openly redistributable reference implementation of the major components of the DNS, including the named process, the DNS resolver library, and tools for verifying the proper operation of the DNS server. The BIND DNS Server is used on the vast majority of name-serving machines on the Internet, providing a robust and stable architecture on top of which an organization's naming architecture can be built.

BIND. See Berkeley Internet Name Domain (BIND).

DNS. See Domain Name System (DNS).

Domain Name System (DNS). A system that defines a hierarchical, yet distributed, database of information about hosts on a network. A domain name is a meaningful and easy-to-remember “handle” for an Internet address. The network administrator configures the DNS with a list of hostnames and Internet protocol (IP) addresses, allowing users of workstations that are configured to query the DNS to specify remote systems by hostnames rather than by IP addresses. DNS domains should not be confused with Windows NT networking domains.

Guardian. An environment available for interactive or programmatic use with the HP NonStop™ Kernel operating system. Processes that run in the Guardian environment usually use the Guardian system procedure calls as their application program interface. Interactive users of the Guardian environment usually use the HP Tandem Advanced Command Language (TACL) or another HP product’s command interpreter. Contrast with Open System Services (OSS).

Incremental Zone Transfer (IXFR). A protocol enhancement to full zone transfer mechanism (AXFR) that transfers only the changed portion(s) of a zone instead of transfering the entire zone file.

Internet protocol (IP). A data communications protocol that handles the routing of data through a network, which typically consists of many different subnetworks. IP is connectionless. It routes data from a source address to a destination address.

Internet protocol version 4 (IPv4). The most widely deployed version of the Internet protocol. IPv4 provides some basic traffic classification mechanisms with its IP Precedence/CBQ and Type of Service header fields. However, network hardware and software have not been configured to use them.

Internet protocol version 6 (IPv6). An update to the Internet protocol version 4 (IPv4). Most of the updates concentrate on basics such as expanding the IP address numbering scheme to accommodate the growth of the Internet. However, IPv6 does include a Class header field that is explicitly intended to designate a Class of Service (CoS), which is an extension of IPv4's IP Precedence/CBQ field.

Internet Systems Consortium (ISC). A nonprofit public benefit corporation dedicated to supporting the infrastructure of the universal connected self-organizing Internet—and the autonomy of its participants—by developing and maintaining core production quality software, protocols, and operations. The *BIND 9 Administrator Reference Manual* is located on the ISC web site (http://www.isc.org).

IP. See Internet protocol (IP).

IPv4. See Internet protocol version 4 (IPv4).

IPv6. See Internet protocol version 6 (IPv6).

ISC. See Internet Systems Consortium (ISC).

IXFR. See Incremental Zone Transfer (IXFR).

named.conf file. The main configuration file of the Name Server that contains server parameters and references to other data files containing the host information.

named process. A daemon that is the core process of DNS. The named process requires a configuration file (named.conf) which will tell where to load the zone files from.

OpenSSL. A toolkit containing implementations of Secure Socket Layer (SSL), Transport Layer Security, and Cryptographic algorithms.

Open System Services (OSS). An open system environment available for interactive or programmatic use with the HP NonStop™ Kernel operating system. Processes that run in the OSS environment usually use the OSS application program interface. Interactive users of the OSS environment usually use the OSS shell for their command interpreter. Synonymous with *Open System Services (OSS) environmen*t. Contrast with Guardian.

OSS. See Open System Services (OSS).

Primary Name Server. A Name Server that reads the zone data from a file on its host.

Request for Comment (RFC). A formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties. Some RFCs are informational in nature. Of those that are intended to become Internet standards, the final version of the RFC becomes the standard and no further comments or changes are permitted. Change can occur, however, through subsequent RFCs that supersede or elaborate on all or parts of previous RFCs.

resolv.conf file. A file that contains details about the address of the Name Server to contact and the domain name to which it belongs.

Resource Record (RR). A DNS data record.

RFC. See Request for Comment (RFC).

RR. See Resource Record (RR).

Simple Network Management Protocol (SNMP). An asynchronous request-response protocol used for network management. SNMP originated as a means for managing TCP/IP and Ethernet networks. Open System Management (OSM) packages can include an SNMP-compliant interface for communication between the system console and HP NonStop™ S-series server.

SNMP. See Simple Network Management Protocol (SNMP).

TCP. See Transmission Control Protocol (TCP).

Transmission Control Protocol (TCP). A connection-oriented protocol that provides for the reliable exchange of data between a sending and a receiving system. TCP implements functions corresponding to the Open Systems Interconnection (OSI) reference model Layer 4, the transport layer.

zone file. A file on a Name Server that designates a domain name along with all of its associated subdomains, IP addresses, and mail server. Parts of a zone file include the A record, CNAME, and MX records.

Index