Chapter 6

BIND Name Server

The Berkeley Internet Name Domain (BIND) server implements the Internet Domain Name Service (DNS) for the IRIX operating system. A name server is a network service that enables clients to name resources or objects in the network and share this information with other network objects. In effect, a name server is a distributed database system for objects in a computer network. All IRIX network programs can use BIND to store and retrieve station names and addresses. You can use BIND to replace the original *host* table lookup of information in the */etc/hosts* file.

BIND has two parts: the name server program, *named,* and a set of C library "resolver" routines that access the server. *named* is a daemon that runs in the background and responds to UDP and TCP queries on a well-known network port. The library routines reside in the standard C library, *libc.a.* The host-address lookup routines *gethostbyname*, *gethostbyaddr*, and *sethostent* use the resolver routines to query the name server. The resolver library routines described in *resolver* include routines that build query packets and exchange them with the name server.

The following topics are covered in detail in this chapter:

- "The Domain Name Service" on page 99
- "BIND Servers and Clients" on page 101
- "The BIND Configuration Files" on page 104
- "Setting Up a BIND Configuration" on page 109
- "Managing the BIND Environment" on page 119
- "Debugging named" on page 121

The Domain Name Service

Host-table lookup routines, such as those using the */etc/hosts* file, require that the master file for the entire network be maintained at a central location by a few people. This approach works well for small networks where there are only a few stations and there is cooperation among the different organizations responsible for them. However, this approach does not work well for large networks where stations cross organizational boundaries.

The Domain Name Service eliminates the need for a single, centralized clearinghouse for all names. The authority for this information can be delegated to the organizations on the network that are responsible for it.

The Domain Name Service is organized as a hierarchical name space, like the IRIX file system. Figure 6-1 shows a small section of this hierarchy. Each subtree in the hierarchy is called a *domain*, and is given a label. At the top of the hierarchy is the root domain, which is labelled with the null label (“ ”). The name of the domain is the concatenation of all the domain labels from the root to the current domain. The labels are listed from right to left and are separated by dots. (Thus the name for the domain labelled “engr” in Figure 6-1 would be “engr.spices.com”.) A label must be unique only within its domain.

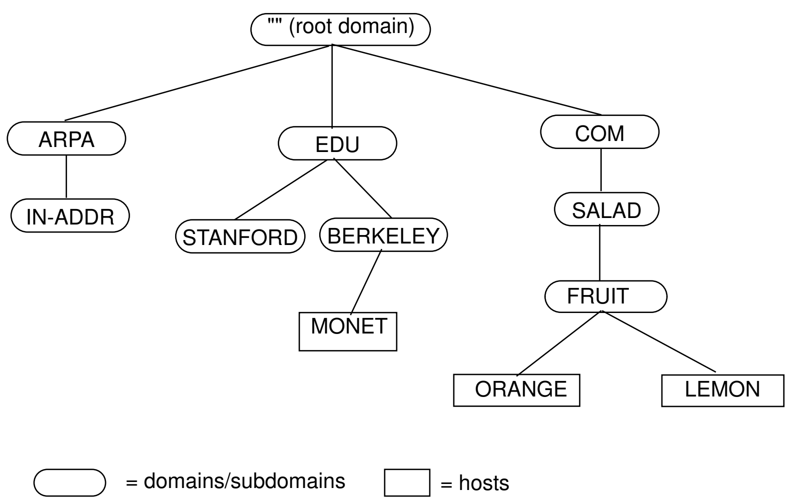

Figure 6-1 Partial View of Domain Name Space

Immediately below the root domain is a set of top-level domains. These top-level domains are relatively static, and are administered by the Network Information Center.

These are the current top-level domains registered with the Network Information Center:

arpa	a temporary domain for stations, also used as the top-level domain for address-to-name mapping
com	companies and businesses
edu	universities and other educational institutions
gov	government agencies
mil	military organizations
net	various network-type organizations and network management-related organizations, such as information centers and operations centers
org	technical support groups, professional societies, or similar organizations

There are also many national domains, such as DE for Germany and FR for France.

The whole space is partitioned into several non-overlapping areas of authority called *zones*. Information in each zone is handled by the zone's "authoritative" or "master" name server(s). Each zone starts at a domain and extends down to the leaf domains, or to domains where other zones start. Zones usually represent administrative boundaries.

An example of a domain name for a station at the University of California, Berkeley is

```
monet.berkeley.edu.
```

The top-level domain for educational organizations is edu. Berkeley is a subdomain of edu, and monet is the name of the station.

BIND Servers and Clients

BIND is based on a server-client relationship. There are several different classes of servers, with varying degrees of authority. This section discusses the interaction between various types of servers, and between servers and clients. This information is summarized in Table 6-1.

Table 6-1 summarizes the general characteristics for various BIND server configurations.

Table 6-1 BIND Server Configurations

Primary Server	**Secondary Server**	**Caching-only Server**	**Forwarder Server**	**Slave Server**
Authoritative server for the domain	"Delegated" authority from primary server	Non-authoritative	Non-authoritative	Non-authoritative
Loads data from local file	Loads data from primary server	Answers queries or forwards queries to authoritative servers	Answers recursive requests or interacts with other name servers before answering requests	Accesses data from specified list of servers (forwarders)

All server configurations must run the *named* server daemon. The *named* daemon is started automatically during station startup if the configuration flag **named** is "on." See the chkconfig(1M) reference page for more details.

The client accesses data from the name servers specified in its *resolv.conf* file. It does not run the domain server, *named*.

Master Servers

A master server for a domain is the authority for that domain. This server maintains all the data corresponding to its domain. Each domain should have at least two master servers: a primary master, and a secondary master to provide backup service if the primary is unavailable or overloaded. A server can be a master for multiple domains, serving as primary for some domains and secondary for others.

A primary master server is a server that loads its data from a file on disk. This server can also delegate authority to other servers in its domain. A secondary master server is a server that is delegated authority and receives its data for a domain from a primary master server. At boot time, the secondary server requests all the data for the given domain from the primary master server. This server then periodically checks with the primary server to see if it needs to update its data.

Root servers are the master servers for the root and top-level Internet domains. They are listed in the *root.cache* file described in "BIND's root.cache File" on page 108.

Slave and Forwarding Servers

A slave server always forwards queries it cannot satisfy locally to a fixed list of forwarding servers, instead of interacting with the master name server for the root and other domains. There may be one or more forwarding servers, and they are tried in turn until the list is exhausted.

A slave-and-forwarder configuration is useful when you do not want all the servers at a given site to interact with the rest of the Internet servers. The stations might be administratively prohibited from having Internet access. To give the stations the appearance of access to the Internet domain system, the stations could be slave servers to the forwarding server on the gateway station. The gateway server would forward the queries and interact with other name servers on the Internet to resolve each query before returning the answer. A benefit of using the forwarding feature is that the central station develops a more complete cache of information, which all the stations can take advantage of. The use of slave mode and forwarding is discussed further in "Setting Up a BIND Configuration" on page 109.

There are two main reasons to use forwarders. First, if your station does not have full network access, it cannot send IP packets to the rest of the network. Therefore, it must rely on a forwarder with access to the network. Second, the forwarder can see all queries as they pass through the server and, therefore, builds up a more complete cache of data than the cache in a typical station name server. In effect, the forwarder becomes a meta-cache from which stations can benefit, thereby reducing the total number of queries from that site to the rest of the network.

Caching-Only Server

A caching-only server is not authoritative for any domain. It services queries and asks other servers, who have the authority, for needed information. The results of queries are cached, to reduce traffic to the authoritative server. Query responses include a *time-to-live* field, which indicates how long they should be cached for.

Clients

A BIND client accesses the name servers that run on other stations in the network. The *named* server does not run on the client station.

The BIND Configuration Files

This section discusses the various BIND configuration files. Examples of these files are provided in "Setting Up a BIND Configuration" on page 109.

In IRIX, the *named* database files are stored in the */var/named* directory. A *README* file contains a short summary of the setup procedure and a list of official names for BIND clients and servers. Typically, servers are also clients. BIND clients require the */etc/resolv.conf* file.

The */var/named/Examples* subdirectory contains sample *named* database files. The files in the *Examples* directory should be used and changed to reflect your setup. These files use the record format described in Appendix A, "BIND Standard Resource Record Format." The database files needed to set up your BIND environment are:

- *named.boot*
- *root.cache*
- *named.hosts*
- *named.rev*
- *localhost.rev*

Note: If your network has more than one domain, incorporate the domain name as part of the *named.hosts*, *named.rev*, and *localhosts.rev* filenames when you create your versions of these files.

The number and configuration of the database files depend on the server type.

Table 6-2 summarizes which database files are required for each type of server.

Table 6-2 named Database Files

Filename	Primary Server	Secondary Server	Caching-Only Server	Forwarder Server	Slave Server
named.boot	required	required	required	required	required
localhosts.rev	required	required	required	required	required
named.hosts	required	N/A	N/A	N/A	N/A
named.rev	required	N/A	N/A	N/A	N/A
root.cache	required	required	required	required	required

BIND's Boot File

The boot file is first read when *named* starts up. It tells the server what type of server it is, which zones it has authority over, and where to get its initial data. The default name of this file is */etc/named.boot.* The template for this file is called */var/named/Examples/named.boot.master* (for primary server) and *named.boot.slave* (for secondary server).

To use a different file, create or modify the */etc/config/named.options* file with this entry:

```
-b other-bootfile-name
```

The recognized boot file structures are described in the subsections that follow.

Directory

The directory line specifies the directory in which the name server should run, allowing the other filenames in the boot file to use relative pathnames.

```
directory /var/named
```

This entry is required. It makes sure *named* is in the proper directory when you try to include files by relative pathnames with $INCLUDE. It also allows *named* to run in a location that is reasonable for dumping core, if necessary.

Primary Master

The line in the boot file that designates a primary server for a zone looks like this:

```
primary   Berkeley.EDU   named.hosts
```

The first field specifies that the server is a primary one for the zone stated in the second field. The third field is the name of the file from which the data is read.

Secondary Master

The line for a secondary server is similar to that for the primary, except that it lists addresses of other servers (usually primary servers) from which the zone data is obtained. For example:

```
secondary  Berkeley.EDU 128.32.0.10 128.32.0.4 ucbhosts.bak
```

The first field specifies that the server is a secondary master server for the zone stated in the second field. The two network addresses specify the name servers that are primary for the zone. The secondary server gets its data across the network from the listed servers. It tries each server in the order listed until it successfully receives the data from a listed server.

If a file name is present after the list of primary servers, data for the zone is saved in that file. When the server first starts, it loads the data from the backup file if possible, and consults a primary server to check that the zone information is still up to date.

Caching-Only Server

All servers should have a line like this one in the boot file to prime the name server's cache:

```
cache   .   root.cache
```

All listed cache files are read when *named* starts up. Valid values are reinstated in the cache, and the root name server information in the cache files is always used to handle initial queries.

The name server needs to know the servers that are the authoritative name servers for the root domain of the network. The *root.cache* file primes the server's cache with the addresses of these higher authorities. This file uses the Standard Resource Record format (or Master File format) described in detail in Appendix F.

You do not need a special line to designate that a server is a caching server. What denotes a caching-only server is the absence of authority lines, such as *secondary* or *primary,* in the boot file.

Forwarders

Any server can make use of forwarders. For example, a server capable of processing recursive queries may try resolving queries on behalf of other stations. The *forwarders* command specifies forwarders by Internet address as follows:

```
forwarders    128.32.0.10     128.32.0.4
```

Slave Mode

You can use slave mode if, because of limited network access, use of forwarders is the only way to resolve queries. You can also use slave mode if you wish to prevent the name server from using forwarders other than those listed. Slave mode is activated by the following command in the boot file:

```
slave
```

If you use *slave,* you must specify forwarders. In slave mode, the server forwards each query to each of the forwarders until an answer is found or the list of forwarders is exhausted.

BIND's named.hosts File

This file contains the host-address database for your domain. It is required for primary servers.

BIND's named.rev File

This file specifies the IN-ADDR.ARPA domain, which is used to translate IP addresses into names. Because Internet addresses do not fall within domain boundaries, this special domain was formed to allow inverse mapping. The IN-ADDR.ARPA domain for a station has four labels preceding it. These labels correspond to the four octets of an Internet address in reverse order. All four octets must be specified, even if an octet is zero.

For example, the Internet address 128.32.130.12 is located in the domain 12.130.32.128.IN-ADDR.ARPA. This reversal of the address allows for the natural grouping of stations in a network.

An IN-ADDR.ARPA domain can also represent a network. For example, if the ARPANET is network 10, there is a domain called 10.IN-ADDR.ARPA.

BIND's localhost.rev File

This file specifies the IN-ADDR.ARPA domain of the local loopback interface's network address, 127.0.0.1. The address is better known as the *localhost* address. Many important network programs depend on the information in this domain. This file is required on all servers.

BIND's root.cache File

This file, by default, contains the initial cache data for root domain servers. It is required, in one form or another, on all servers.

BIND's /etc/config/named.options File

This file is optional. It is used during station startup and by the *named.restart* script. Specify command-line arguments for *named* in this file. See the named(1M) reference page for details on the options.

Configuring Hostname Resolution With /etc/resolv.conf

The only configuration file required by BIND clients is the */etc/resolv.conf* file.This file is read the first time *gethostbyname* or *gethostbyaddr* is called. The *resolv.conf* file has several functions:

- It defines the default domain or the default domain search list.
- It specifies the ordering of host resolution services used by *gethostbyname* and *gethostbyaddr*.
- It lists Internet addresses of name servers.

The first two items apply to both client and server stations. The last item is required only by client stations. The file's format is described in detail in the resolver(4) reference page.

To set up a station as a client of remote servers, add **nameserver** entries for the Internet addresses of the name servers to */etc/resolv.conf.* For example:

```
nameserver 128.32.130.12
```

You can specify up to three *nameserver* entries. It is usually not necessary to create this file if you have a local server running. An entry for the local server should use an Internet address of 0 (meaning "this station").

On client and server stations, the name in */etc/sys_id* should be set to the fully qualified domain name. For example:

```
monet.Berkeley.EDU
```

However, if you choose not to use fully qualified domain names, add a line with the keyword *domain* and the station's domain to the *resolv.conf* file. For example:

```
domain berkeley.edu
```

The *gethostbyname* and *gethostbyaddr* library routines are normally configured to access station information in this order:

1. NIS
2. BIND
3. Local */etc/hosts* file

You can change this behavior with the *hostresorder* keyword in */etc/resolv.conf.* See the resolver(4) reference page for details.

To enable the system manager to copy files from another station when it is in single-user mode, the */etc/hosts* file should contain entries for important stations in addition to the entries for the local station's network interface(s) and *localhost*. See the hosts(4) reference page for more information about the format.

Setting Up a BIND Configuration

This section provides an example of how a BIND environment might be organized and describes the procedure for configuring the various servers and client stations. The example assumes you are connected to the Internet. When setting up your own environment, replace the variables in the example with your own BIND environment variables. The example is based on these variables:

- the domain is *fruit.com*, network address 192.35.10, and the network is attached to the Internet
- primary server is *apples.fruit.com*, internet address 192.35.10.1
- aecondary server is *oranges.fruit.com*, internet address 192.35.10.2
- dorwarding server is *banana.fruit.com*, internet address 192.35.10.3
- caching-only server is *guava.fruit.com*, internet address 192.35.10.4
- slave servers are *pineapple1.fruit.com* and *pineapple2.fruit.com*, Internet addresses 192.35.10.8 and 192.35.10.9
- clients are *plum1.fruit.com, plum2.fruit.com*, and *plum3.fruit.com*, Internet addresses 192.35.10.5, 192.35.10.6, and 192.35.10.7

Figure 6-2 illustrates the example BIND environment described above. Station names in the figure are shortened for illustrative purposes only.

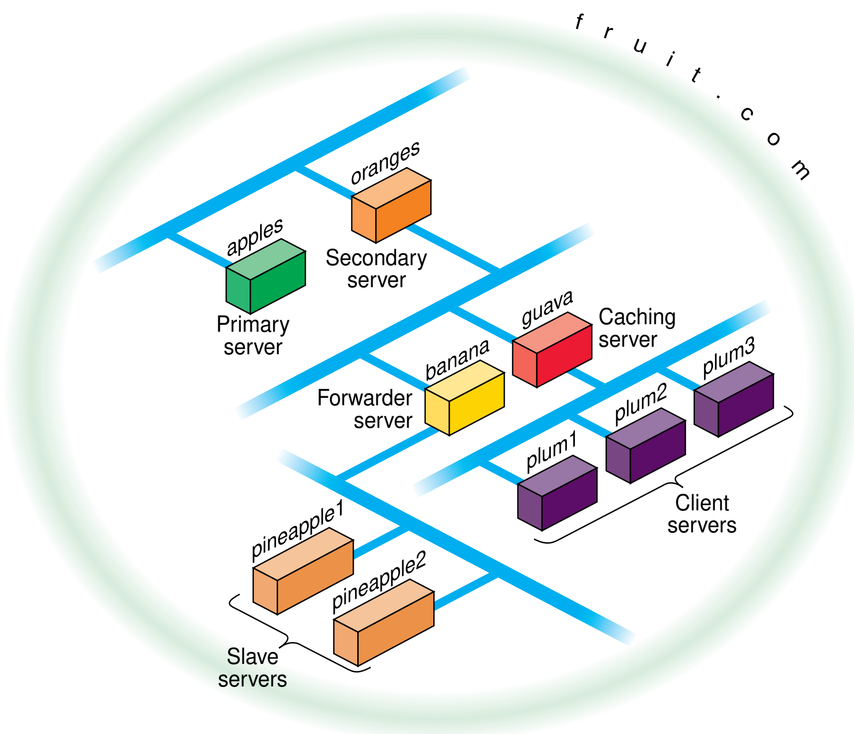

Figure 6-2 Example BIND Configuration

Configuring the Primary Server

Use this procedure to configure a primary server:

1. Log in as *root*.
2. Move to the named example directory:

   ```
   cd /var/named/Examples
   ```

3. Copy the template files to the */var/named* directory:

   ```
   cp named.boot.master root.cache named.hosts \
   named.rev localhost.rev /var/named
   ```

4. Move *named.boot.master* to the default filename:

```
cd ..
mv named.boot.master named.boot
```

5. Modify *named.boot* with your editor of choice to resemble the following:

```
;
; Boot file for apple.fruit.com, primary for fruit.com
;
directory /var/named
;type    domain     source host/file     backup file
cache    .          root.cache
primary  fruit.com  fruit.named.hosts
```

6. Modify the *named.hosts* file, here called *fruit.named.hosts*, to resemble the following:

```
; Authoritative data for fruit.com
;
@    IN  SOA  apples.fruit.com. named-mgr.apples.fruit.com.
            (1994021501 ; Serial
             10800      ; Refresh 3 hours
             3600       ; Retry 1 hour
             3600000    ; Expire  1000 hours
             86400 )    ; Minimum 24 hours
; authoritative name servers for fruit.com
          IN    NS     apples.fruit.com.
          IN    NS     oranges.fruit.com.
; address records for all hosts on the net
          IN     A     192.35.10.1
apples    IN     A     192.35.10.1
oranges   IN     A     192.35.10.2
banana    IN     A     192.35.10.3
guava     IN     A     192.35.10.4
plum1     IN     A     192.35.10.5
plum2     IN     A     192.35.10.6
plum3     IN     A     192.35.10.7
pineapple1 IN     A     192.35.10.8
pineapple2 IN     A     192.35.10.9
localhost IN     A     127.0.0.1
; canonical or alias name for localhost
loghost   IN        CNAME localhost
```

7. Modify the *localhost.rev* file to resemble the following:

```
;localhost.rev  -- PTR record for 127.1
;
@  IN  SOA  apples.fruit.com. named-mgr.apples.fruit.com.
```

```
                              (1994021501 ;Serial
                               10800      ;Refresh 3 hours
                               3600       ;Retry   1 hour
                               3600000    ;Expire 1000 hours
                               86400 )    ;Minimum 24 hours
; authoritative name servers for fruit.com
   IN      NS      apples.fruit.com.
   IN      NS      oranges.fruit.com.
0  IN      PTR     loopback.fruit.com.
1  IN      PTR     localhost.
```

8. Modify the *named.rev file* (*fruitnamed.rev*) to resemble the following:

```
;
;       @(#)named.rev   1.1     (Berkeley)      86/02/05
;
@  IN  SOA  apples.fruit.com. named-mgr.apples.fruit.com.
                               (1994021501 ; Serial
                                10800 ; Refresh 3 hours
                                3600  ; Retry   1 hour
                                3600000 ; Expire  1000 hours
                                86400 ); Minimum 24 hours
;authoritative name servers for fruit.com
              IN      NS      apples.fruit.com.
              IN      NS      oranges.fruit.com.
;named.rev addresses, by default, are the last two numbers
;of the internet addresses in reverse order, if Class B
;address.  If Class C address, then it's the last number.
1             IN    PTR      apples.fruit.com.
2             IN    PTR      oranges.fruit.com.
3             IN    PTR      banana.fruit.com.
4             IN    PTR      guava.fruit.com.
5             IN    PTR      plum1.fruit.com.
6             IN    PTR      plum2.fruit.com.
7            IN    PTR    plum3.fruit.com.
8            IN    PTR    pineapple1.fruit.com.
9            IN    PTR    pineapple2.fruit.com.
```

9. Use the default *root.cache* file if the primary server is attached to the Internet. If practical, you should obtain the most up-to-date list from rs.internic.net using anonymous FTP. The list is kept in the file *domain/named.root*.

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
```

```
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    Sep 1, 1995
;       related version of root zone:   1995090100
;
;
; formerly NS.INTERNIC.NET
;
.                         3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.       3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                         3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.       3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                         3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.       3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                         3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.       3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                         3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.       3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                         3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.       3600000      A     39.13.229.241
;
; formerly NS.NIC.DDN.MIL
```

```
;
.                         3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.       3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                         3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.       3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                         3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.       3600000      A     192.36.148.17
; End of File
```

10. Enable *named* and reboot the station with the following commands:

    chkconfig named on
    reboot

Configuring the Secondary Server

Use this procedure to configure a secondary server:

1. Log in as *root*.
2. Move to the *named* example directory:

   cd /var/named/Examples

3. Copy the template files to the */var/named* directory:

   cp named.boot.slave root.cache localhost.rev /var/named

4. Move *named.boot.slave* to the default filename:

   cd ..

   mv named.boot.slave named.boot

5. Modify *named.boot* to look like the following:

   more named.boot

```
;
; Boot file for orange.fruit.com, secondary for fruit.com
;
directory  /var/named
```

```
; type      domain      source host/file   backup file
cache       .           root.cache
secondary   fruit.com   192.35.10.1        fruithosts.bak
```

6. Use the same *localhost.rev* file you installed on your primary server.
7. Use the same *root.cache* file you installed on your primary server.
8. Enable *named* and reboot the station with the following commands:

   chkconfig named on

   reboot

Configuring a Caching-Only Server

Use this procedure to set up a caching-only server:

1. Log in as *root*.
2. Move to the *named* example directory:

   cd /var/named/Examples
3. Copy the template files to the */var/named* directory:

   cp named.boot.master root.cache /var/named
4. Move *named.boot.master* to the default filename:

   cd ..

   mv named.boot.master named.boot
5. Modify *named.boot* to look like the following:

   more named.boot

```
;
;Boot file for guava.fruit.com,caching-only server for
;fruit.com
;Note that there should be one primary entry for each SOA
;record.
;
;
directory          /var/named
;type    domain   source host/file  backup file
cache    .        root.cache
```

6. Use the same *localhost.rev* file you installed on your primary server.

7. Use the same *root.cache* file you installed on your primary server.
8. Enable *named* and reboot the station with the following commands:

   chkconfig named on

   reboot

Configuring the Forwarding Server

Use this procedure to set up a forwarding server:

1. Log in as *root*.
2. Move to the *named* example directory:

   cd /var/named/Examples
3. Copy the template files to the */var/named* directory:

   cp named.boot.master root.cache localhost.rev /var/named
4. Move *named.boot.master* to the default filename:

   cd ..

   mv named.boot.master named.boot
5. Modify *named.boot* to look like the following:

   more named.boot

```
;
;Boot file for banana.fruit.com, forwarder server
;for fruit.com
;Note that there should be one primary entry for each
;SOA record.
;
;
directory       /var/named

;type       domain        source host/file   backup file
cache       .             root.cache
forwarders  192.35.10.1   192.35.10.2
```

6. Use the same *localhost.rev* file you installed on your primary server.
7. Use the same *root.cache* file you installed on your primary server.
8. Enable *named* and reboot the station with the following commands:

chkconfig named on

reboot

Configuring a Slave Server

1. Log in as *root*.
2. Move to the *named* example directory:

   cd /var/named/Examples

3. Copy the template files to the */var/named* directory:

   cp named.boot.slave root.cache localhost.rev /var/named

4. Move *named.boot.master* to the default filename:

   cd ..

   mv named.boot.slave named.boot

5. Modify *named.boot* to look like the following:

```
;
;Boot file for pineapple1.fruit.com, slave server for
;fruit.com
;
directory       /var/named
;type       domain      source host/file   backup file
cache       .           root.cache
forwarders  192.35.10.3
slave
```

6. Use the same *localhost.rev* file you installed on your primary server.
7. Use the same *root.cache* file you installed on your primary server.
8. Enable *named* and reboot the station with the following commands:

   chkconfig named on

   reboot

Configuring the Client

Use this procedure to set up a BIND client:

1. Log in as *root*.
2. Create or modify the *resolv.conf* file to include the default domain name, the host resolution order, and the list of name servers. It should look something like this:

```
domain fruit.com
nameserver 192.35.10.4
nameserver 192.35.10.2
nameserver 192.35.10.1
hostresorder bind local
```

3. Rebooting the client is suggested, but not required.

Managing the BIND Environment

This section describes the steps involved in maintaining the databases. It details how to add and delete a station from the domain and how to add a new subdomain. It also discusses some of the scripts that manage the BIND database.

Adding a New Station

To add a new station to your zone files:

1. Edit the appropriate zone file for the station's domain.
2. Add an *A* record for each address of the station.
3. Add *CNAME*, *HINFO*, *WKS*, *RP,* and *MX* records (optional).
4. Add the reverse IN-ADDR entry for each station address in the appropriate zone files for each network the station is on.

Deleting a Station

To delete a station from the zone files:

1. Remove all the station's resource records from the zone file of the station's domain.
2. Remove all the station's *PTR* records from the IN-ADDR zone files for each network the station was on.

Adding Another Domain

To add a new subdomain to your domain:

1. Set up the other domain server, the new zone file, or both.
2. For each server of the new domain, add an *NS* record to the zone file of the parent domain.
3. Add any necessary glue address records. See Appendix A, “BIND Standard Resource Record Format” for details about glue records.

Management Scripts

You can use two shell scripts to manage *named*: */usr/sbin/named.reload* and */usr/sbin/named.restart*.

named Reload Script

This shell script sends the HUP signal to *named,* which causes it to read *named.boot* and reload the database. All previously cached data is lost. Use this script when *named* is running and you want the internal database for *named* to reflect any changes you have made.

named Restart Script

This shell script terminates the running *named* and starts a new one. Use this script when you have made changes to the *named.boot* file, or whenever you need to place the server in a known state.

Debugging named

When *named* is running incorrectly, first check */var/adm/SYSLOG* for any messages. For additional information, send *named* one of the following signals, using *killall*(1M) and defining SIG as INT, ABRT, USR1, or USR2:

```
/etc/killall -SIG named
```

INT	Dumps the current database and cache to */var/tmp/named_dump.db.* This dumping should indicate whether the database was loaded correctly.
ABRT	Dumps statistics data into */var/tmp/named.stats.* Statistics data is appended to the file.
USR1	Turns on debugging. Each subsequent USR1 increases the debug level. There are 10 debug levels, and each prints more detailed information. A debug level of 5 is useful for debugging lookup requests. The output goes to */var/tmp/named.run.*
USR2	Turns off debugging completely.

SYSLOG Messages

Using *syslog*(3B), *named* logs certain errors to */var/adm/SYSLOG*. This section lists important error messages and their meanings.

- *dname* `has CNAME and other illegal data`

  An alias has more than just a *CNAME* record. For example, since only “monet” should have the *A* record, the following is wrong:

  ```
  ucbmonet IN CNAME monet
  ucbmonet IN A 128.32.0.1
  ```

- `Attempted to query myself on ipaddr as name server for` *dname*

  The station is listed incorrectly as a forwarder in the *named.boot* file.

- `zoneref: Masters for secondary zone` *dname* `unreachable`

  This station is a secondary server for *dname*. The primary server returned an invalid data response or, because of a network problem, the station was not able to contact the primary server to obtain the current state of the zone information.

- `Lame delegation to` *dname1* `received from` *ipaddr* `(purported server for` *dname2*`) on query on name [`*dname3*`]`

The message indicates that the remote server at the specified address is supposed to be authoritative for the *dname2* domain but has returned an indication that implies it is not. The remote server or the server for its parent domain is misconfigured. This error message can be disabled if *named* is started with the **-L lamedel** option.

- `MAXQUERIES exceeded, possible data loop in resolving` *dname*

  The name server has tried to query too many other servers for the specified record. This might happen if each of two remote servers reply that the other has the desired information for *dname*.

- `Malformed response from` *ipaddr*

  The remote DNS server at the specified address returned a malformed packet. This message typically indicates an error in the remote server.

- `Bogus root NS` *dname1* `received from` *ipaddr* `on query on name[`*dname2*`] -- rejected`

  The name server at the specified address improperly returned NS records for the root domain. The records have been ignored. You can disable this error message if *named* is started with **-L rootns.**

- `Root NS` *dname1* `received from` *ipaddr* `on query on name [`*dname2*`]`

  The name server at the specified address returned NS records for the root domain. You can disable this informational message if *named* is started with the **-L rootns** option.

The nslookup Command

The *nslookup* command is a useful debugging tool for querying local and remote name servers. *nslookup* operates in either interactive or non-interactive mode. Interactive mode allows the user to query the name server for information about various stations and domains or print a list of stations in the domain. Non-interactive mode is used to print just the name and requested information for a station or domain. The following example of *nslookup* output gets the address record for the station monet.berkeley.edu:

```
Default Server:  ucbvax.berkeley.edu
Address:  128.32.133.1
> monet
Server:  ucbvax.berkeley.edu
Address:  128.32.133.1
Name:    monet.berkeley.edu
Address:  128.32.130.6
```

To exit, press **`<Ctrl-D>`** or enter **`exit`**. The *help* command summarizes available commands. The complete set of commands is described on the nslookup(1C) reference page.