Appendix A

BIND Standard Resource Record Format

The Berkeley Internet Name Domain (BIND) server uses a specific record format for the name server data files. This appendix details BIND's standard resource record format by resource record type.

Standard Resource Record Format

The records in the name server data files are called *resource records*. The Standard Resource Record (RR) Format is specified in RFC 1035. The standard format of resource records is:

```
{name} {ttl} addr-class Record Type Record-specific data
```

- The first field is the name of the domain record. It must always start in column 1. For some RRs the name may be left blank, in which case it becomes the name of the previous RR.
- The second field is an optional time-to-live field, which specifies how long this data will be stored in the database. When this field is blank, the default time-to-live value is specified in the Start of Authority (SOA) resource record (described later in this section).
- The third field is the address class. Currently only the *IN* class (for Internet hosts and addresses) is recognized.
- The fourth field identifies the type of resource record.
- Subsequent fields depend on the type of RR.

Case is preserved in names and data fields when they are loaded into the name server. Comparisons and lookups in the name server database are not case sensitive.

If you specify TTLs for resource records, it is important that you set them to appropriate values. The TTL is the amount of time (in seconds) that a resolver will use the data from your server before it asks your server again. If you set the value too low, your server will

become loaded down with repeat requests. If you set it too high, information you change will not be distributed in a reasonable amount of time.

Most host information does not change much over time. A good way to set up your TTLs is to set them at a high value, and lower the value if you know a change is coming soon. You might set most TTLs between a day (86400) and a week (604800). When you know some data is changing soon, set the TTL for that RR to a low value, between an hour (3600) and a day, until the change takes place. Then reset it to its previous value. All resource records with the same name, class, and type should have the same TTL value.

The following characters have special meanings in resource records:

(blank)	A blank or tab character in the name field denotes the current domain.
@	A free-standing "at" sign (@) in the name field denotes the current origin.
.	A free-standing period in the name field represents the root domain name.
\x	The backslash designates that the special meaning of the character x does not apply. The x represents any character other than a digit (0–9). For example, use \.to place a dot character in a label.
\DDD	Each D is a digit; the complete string is the octet corresponding to the decimal number described by DDD. The octet is assumed to be text and is not checked for special meaning.
( )	Parentheses enclose group data that crosses a line. In effect, newlines are not recognized within parentheses. This notation is useful with SOA and WKS records.
;	A semicolon precedes a comment; the remainder of the line is ignored.
*	An asterisk is a wildcard character.

Usually a resource record has the current origin appended to the name if the name is not terminated by a period (.). This scheme is useful for appending the current domain name to the data, such as workstation names, but can cause problems if you do not want the name to be appended. If the name is not in the domain for which you are creating the data file, end the name with a period. However, do not append the period to Internet addresses.

$INCLUDE

An include line has $INCLUDE starting in column 1 and is followed by a filename. This feature helps you use multiple files for different types of data. For example:

```
$INCLUDE /usr/etc/named.d/mailboxes
```

This line is a request to load the file */usr/etc/named.d/mailboxes*. The $INCLUDE command does not cause data to be loaded into a different zone or tree. It allows data for a given zone to be organized in separate files. For example, you might keep mailbox data separate from host data by using this mechanism.

$ORIGIN

$ORIGIN changes the origin in a data file. The line starts in column 1 and is followed by a domain origin. This feature is useful for putting more than one domain in a data file.

```
$ORIGIN Berkeley.EDU.
```

SOA—Start of Authority

```
name {ttl} addr-class SOA Source                Person-in-charge
@          IN          SOA ucbvax.Berkeley.EDU  kjd.ucbvax.Berkeley.EDU.
                       (
                       1994021501;Serial
                       10800   ;Refresh
                       3600    ;Retry
                       3600000 ;Expire
                       86400   ;Minimum
                       )
```

The Start of Authority record, SOA, designates the start of a zone. There should be only one SOA record per zone.

The name is the name of the zone. It can be a complete domain name like "Berkeley.EDU." or a name relative to the current $ORIGIN. The "at" sign (@) indicates the current zone name, taken from the "primary" line in the *named.boot* file or from a previous $ORIGIN line.

Source is the name of the host on which the master data file resides, typically the primary master server.

Person-in-charge is the mailing address for the person responsible for the name server. The mailing address is encoded in the form of a domain name where the "at" sign (@) separating the user name from the hostname is replaced with a period. In the example above, *kjd.ucbvax.berkeley.edu* is the encoded form of *kjd@ucbvax.berkeley.edu*.

Serial is the version number of this data file, and should be incremented whenever data are changed. Do not use floating point numbers (numbers with a decimal point, such as 1.1). A useful convention is to encode the current date in the serial number. For example, *25 April 1994 edit #1* is encoded as:

```
1994042501
```

Increment the edit number if you modify the file more than once on a given day.

Refresh indicates how often, in seconds, a secondary name server is to check with the primary name server to see if an update is needed.

Retry indicates how long, in seconds, a secondary server is to retry after a failure to check for a refresh.

Expire is the maximum number of seconds that a secondary name server has to use the data before they expire for lack of getting a refresh.

Minimum is the default number of seconds to be used for the time-to-live field on resource records with no explicit time-to-live value.

NS—Name Server

```
{name}   {ttl}   addr-class   NS   Name server's_name
                 IN           NS   ucbarpa.Berkeley.EDU.
```

The Name Server record, NS, lists the name of a machine that provides domain service for a particular domain. The name associated with the RR is the domain name, and the data portion is the name of a host that provides the service. Workstations providing name service need not be located in the *named* domain. There should be one NS record for each master server (primary or secondary) for the domain. If you use more than approximately 10 to 15 NS records for a zone, you may exceed DNS datagram size limits.

NS records for a domain must exist in both the zone that delegates the domain and in the domain itself. If the name server host for a particular domain is itself inside the domain, then a glue record is needed. A glue record is an Address (A) record that specifies the

address of the server. Glue records are needed only in the server delegating the domain, not in the domain itself. For example, if the name server for domain SRI.COM is KL.SRI.COM, then the NS and glue A records on the delegating server look like this:

```
SRI.COM.     IN   NS      KL.SRI.COM.
KL.SRI.COM. IN   A       10.1.0.2
```

The administrators of the delegating and delegated domains should ensure that the NS and glue A records are consistent and remain so.

A—Address

```
{name}    {ttl}    addr-class   A      address
ucbvax             IN           A      128.32.133.1
                   IN           A      128.32.130.12
```

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

HINFO—Host Information

```
{name}  {ttl}  addr-class HINFO  Hardware          OS
               IN         HINFO  SGI-IRIS-INDY     IRIX
```

The Host Information resource record, HINFO, is for host-specific data. This record lists the hardware and operating system running at the listed host. Only a single space separates the hardware information and the operating-system information. To include a space in the workstation name, you must place quotation marks around the name. There should be one HINFO record for each host. See the file */usr/etc/named.d/README* for the current list of names for IRIS4D Series workstations and servers. To learn the names for other types of hardware and operating systems, refer to the most current "Assigned Numbers" RFC (RFC 1340 as of this writing).

WKS—Well-Known Services

```
{name} {ttl} addr-class WKS address      protocol services list
             IN         WKS 192.12.6.16  UDP    (who route
                                                  timed domain)
             IN         WKS 192.12.63.16 TCP    (echo telnet
                                                  chargen ftp
```

```
                                                    smtp time
                                                    domain bootp
                                                    finger sunrpc)
```

The Well-Known Services record, WKS, describes well-known services supported by a particular protocol at a specified address. The list of services and port numbers comes from the list of services specified in */etc/services.* There should be only one WKS record per protocol per address.

CNAME—Canonical Name

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

The Canonical Name resource record, CNAME, specifies an alias for the official, or canonical, hostname. This record should be the only one associated with the alias name. All other resource records should be associated with the canonical name, not with the alias. Any resource records that include a domain name as their value (such as NS or MX) should list the canonical name, not the alias.

Aliases are also useful when a host changes its name. In that case, it is usually a good idea to have a CNAME record so that people still using the old name get to the right place.

PTR – Domain Name Pointer

```
name   {ttl}   addr-class   PTR   real name
6.130          IN            PTR   monet.Berkeley.EDU.
```

A Domain Name Pointer record, PTR, allows special names to point to some other location in the domain. The example of a PTR record given here is used to set up reverse pointers for the special IN-ADDR.ARPA domain. PTR names should be unique to the zone. Note the period (.) appended to the real name to prevent *named* from appending the current domain name.

MB—Mailbox

```
name   {ttl}   addr-class   MB   Machine
ben            IN            MB   franklin.Berkeley.EDU.
```

The Mailbox record, MB, lists the workstation where a user receives mail. The *name* field is the user's login. The machine field lists the workstation to which mail is to be delivered. Mailbox names should be unique to the zone.

MR—Mail Rename Name

```
name {ttl} addr-class MR  corresponding_MB
Postmaster IN         MR  ben
```

The Mail Rename Name record, MR, lists aliases for a user. The *name* field lists the alias for the name listed in the last field, which should have a corresponding MB record.

MINFO—Mail Information

```
name  {ttl}  addr-class MINFO  requests      maintainer
BIND         IN         MINFO  BIND-REQUEST  kjd.Berkeley.EDU
```

The Mail Information record, MINFO, creates a mail group for a mailing list. This resource record is usually associated with a Mail Group (MG) record, but can be used with a Mailbox (MB) record. The *name* is the name of the mailbox. The *requests* field is where mail (such as requests to be added to a mail group) should be sent. The *maintainer* is a mailbox that should receive error messages. This arrangement is appropriate for mailing lists when errors in members' names should be reported to someone other than the sender.

MG—Mail Group Member

```
{mail group name}  {ttl}    addr-class  MG  member name
                            IN          MG  Bloom
```

The Mail Group record, MG, lists members of a mail group. Here is an example for setting up a mailing list:

```
Bind  IN  MINFO    Bind-Request        kjd.Berkeley.EDU.
      IN  MG       Ralph.Berkeley.EDU.
      IN  MG       Zhou.Berkeley.EDU.
```

MX—Mail Exchanger

```
                              preference  mail
```

```
name {ttl}      addr-class MX  value       exchanger
Munnari.OZ.AU.  IN         MX  10          Seismo.CSS.GOV.
*.IL.           IN         MX  10          CUNYVM.CUNY.EDU.
```

The Mail Exchanger record, MX, specifies a workstation that can deliver mail to a workstation not directly connected to the network. In the first example given here, Seismo.CSS.GOV. is a mail gateway that can deliver mail to Munnari.OZ.AU. Other systems on the network cannot deliver mail directly to Munnari. The two systems, Seismo and Munnari, can have a private connection or use a different transport medium. The preference value is the order that a mailer should follow when there is more then one way to deliver mail to a single workstation. See RFC 974 for more detailed information.

You can use a wildcard name containing an asterisk (*) for mail routing with an MX record. Servers on the network can state that any mail to a given domain is to be routed through a relay. In the second example given here, all mail to hosts in the domain IL is routed through CUNYVM.CUNY.EDU. This routing is done by means of a wildcard MX resource record, which states that *.IL has an MX of CUNYVM.CUNY.EDU.

RP—Responsible Person

```
owner {ttl} addr RP mbox_domain_name         TXT_domain_name
franklin    IN   RP franklin.berkeley.edu  admin.berkeley.edu.
```

The Responsible Person record, RP, identifies the name or group name of the responsible person for a host. Often it is desirable to be able to identify the responsible entity for a particular host. Otherwise, when that host is down or malfunctioning, it is difficult to contact someone who can resolve the problem or repair the host.

The *mbox domain name* field is a domain name that specifies the mailbox for the responsible person. Its format in master files uses the DNS convention for mailbox encoding, identical to that used for the *Person-in-charge* mailbox field in the SOA record. In the example given here, the *mbox domain name* shows the encoding for ben@franklin.berkeley.edu. You can specify the root domain name (just ".") to indicate that no mailbox is available.

The last field is a domain name for which TXT RRs exist. You can perform a subsequent query to retrieve the associated TXT resource records at the TXT domain name. Retrieving the TXT records provides a level of indirection so that the entity can be referred to from multiple places in the DNS. You can specify the root domain name (just ".") for the TXT domain name to indicate that no associated TXT RR exists. In the

example, “sysadmins.berkeley.edu.” is the name of a TXT record that could contain some text with names and phone numbers.

The format of the RP record is class insensitive. Multiple RP records at a single name may be present in the database. They should have identical TTLs.

The RP record is experimental; not all DNS servers implement or recognize it.

TXT—Text

```
text-name    {ttl}    addr-class    TXT   text-data
location              IN            TXT   "Berkeley, CA"
```

The Text record, TXT, is used to hold descriptive text. The semantics of the text depend on the domain where it is found.