

## DOMAIN ADMINISTRATORS OPERATIONS GUIDE

### STATUS OF THIS MEMO

This RFC provides guidelines for domain administrators in operating a domain server and maintaining their portion of the hierarchical database. Familiarity with the domain system is assumed. Distribution of this memo is unlimited.

### ACKNOWLEDGMENTS

This memo is a formatted collection of notes and excerpts from the references listed at the end of this document. Of particular mention are Paul Mockapetris and Kevin Dunlap.

### INTRODUCTION

A domain server requires a few files to get started. It will normally have some number of boot/startup files (also known as the "safety belt" files). One section will contain a list of possible root servers that the server will use to find the up-to-date list of root servers. Another section will list the zone files to be loaded into the server for your local domain information. A zone file typically contains all the data for a particular domain. This guide describes the data formats that can be used in zone files and suggested parameters to use for certain fields. If you are attempting to do anything advanced or tricky, consult the appropriate domain RFC's for more details.

Note: Each implementation of domain software may require different files. Zone files are standardized but some servers may require other startup files. See the appropriate documentation that comes with your software. See the appendix for some specific examples.

### ZONES

A zone defines the contents of a contiguous section of the domain space, usually bounded by administrative boundaries. There will typically be a separate data file for each zone. The data contained in a zone file is composed of entries called Resource Records (RRs).

You may only put data in your domain server that you are authoritative for. You must not add entries for domains other than your own (except for the special case of "glue records").

A domain server will probably read a file on start-up that lists the zones it should load into its database. The format of this file is not standardized and is different for most domain server implementations. For each zone it will normally contain the domain name of the zone and the file name that contains the data to load for the zone.

## ROOT SERVERS

A resolver will need to find the root servers when it first starts. When the resolver boots, it will typically read a list of possible root servers from a file.

The resolver will cycle through the list trying to contact each one. When it finds a root server, it will ask it for the current list of root servers. It will then discard the list of root servers it read from the data file and replace it with the current list it received.

Root servers will not change very often. You can get the names of current root servers from the NIC.

FTP the file NETINFO:ROOT-SERVERS.TXT or send a mail request to NIC@SRI-NIC.ARPA.

As of this date (June 1987) they are:

SRI-NIC.ARPA	10.0.0.51	26.0.0.73	
C.ISI.EDU	10.0.0.52		
BRL-AOS.ARPA	192.5.25.82	192.5.22.82	128.20.1.2
A.ISI.EDU	26.3.0.103		

## RESOURCE RECORDS

Records in the zone data files are called resource records (RRs). They are specified in RFC-883 and RFC-973. An RR has a standard format as shown:

```
<name>    [<ttl>]    [<class>]    <type>    <data>
```

The record is divided into fields which are separated by white space.

<name>

The name field defines what domain name applies to the given

RR. In some cases the name field can be left blank and it will default to the name field of the previous RR.

#### <ttl>

TTL stands for Time To Live. It specifies how long a domain resolver should cache the RR before it throws it out and asks a domain server again. See the section on TTL's. If you leave the TTL field blank it will default to the minimum time specified in the SOA record (described later).

#### <class>

The class field specifies the protocol group. If left blank it will default to the last class specified.

#### <type>

The type field specifies what type of data is in the RR. See the section on types.

#### <data>

The data field is defined differently for each type and class of data. Popular RR data formats are described later.

The domain system does not guarantee to preserve the order of resource records. Listing RRs (such as multiple address records) in a certain order does not guarantee they will be used in that order.

Case is preserved in names and data fields when loaded into the name server. All comparisons and lookups in the name server are case insensitive.

Parenthesis ("(",")") are used to group data that crosses a line boundary.

A semicolon (";") starts a comment; the remainder of the line is ignored.

The asterisk ("\*") is used for wildcarding.

The at-sign ("@") denotes the current default domain name.

## NAMES

A domain name is a sequence of labels separated by dots.

Domain names in the zone files can be one of two types, either absolute or relative. An absolute name is the fully qualified domain name and is terminated with a period. A relative name does not terminate with a period, and the current default domain is appended to it. The default domain is usually the name of the domain that was specified in the boot file that loads each zone.

The domain system allows a label to contain any 8-bit character. Although the domain system has no restrictions, other protocols such as SMTP do have name restrictions. Because of other protocol restrictions, only the following characters are recommended for use in a host name (besides the dot separator):

"A-Z", "a-z", "0-9", dash and underscore

## TTL's (Time To Live)

It is important that TTLs are set to appropriate values. The TTL is the time (in seconds) that a resolver will use the data it got from your server before it asks your server again. If you set the value too low, your server will get loaded down with lots of repeat requests. If you set it too high, then information you change will not get distributed in a reasonable amount of time. If you leave the TTL field blank, it will default to what is specified in the SOA record for the zone.

Most host information does not change much over long time periods. A good way to set up your TTLs would be to set them at a high value, and then lower the value if you know a change will be coming soon. You might set most TTLs to anywhere between a day (86400) and a week (604800). Then, if you know some data will be changing in the near future, set the TTL for that RR down to a lower value (an hour to a day) until the change takes place, and then put it back up to its previous value.

Also, all RRs with the same name, class, and type should have the same TTL value.

## CLASSES

The domain system was designed to be protocol independent. The class field is used to identify the protocol group that each RR is in.

The class of interest to people using TCP/IP software is the class

"Internet". Its standard designation is "IN".

A zone file should only contain RRs of the same class.

## TYPES

There are many defined RR types. For a complete list, see the domain specification RFCs. Here is a list of current commonly used types. The data for each type is described in the data section.

Designation	Description
=====	
SOA	Start Of Authority
NS	Name Server
A	Internet Address
CNAME	Canonical Name (nickname pointer)
HINFO	Host Information
WKS	Well Known Services
MX	Mail Exchanger
PTR	Pointer

## SOA (Start Of Authority)

```
<name> [<ttl>] [<class>] SOA <origin> <person> (
    <serial>
    <refresh>
    <retry>
    <expire>
    <minimum> )
```

The Start Of Authority record designates the start of a zone. The zone ends at the next SOA record.

<name> is the name of the zone.

<origin> is the name of the host on which the master zone file resides.

<person> is a mailbox for the person responsible for the zone. It is formatted like a mailing address but the at-sign that normally separates the user from the host name is replaced with a dot.

<serial> is the version number of the zone file. It should be incremented anytime a change is made to data in the zone.

<refresh> is how long, in seconds, a secondary name server is to check with the primary name server to see if an update is needed. A good value here would be one hour (3600).

<retry> is how long, in seconds, a secondary name server is to retry after a failure to check for a refresh. A good value here would be 10 minutes (600).

<expire> is the upper limit, in seconds, that a secondary name server is to use the data before it expires for lack of getting a refresh. You want this to be rather large, and a nice value is 3600000, about 42 days.

<minimum> is the minimum number of seconds to be used for TTL values in RRs. A minimum of at least a day is a good value here (86400).

There should only be one SOA record per zone. A sample SOA record would look something like:

```

      @      IN      SOA      SRI-NIC.ARPA.  HOSTMASTER.SRI-NIC.ARPA. (
                                45              ;serial
                                3600            ;refresh
                                600             ;retry
                                3600000        ;expire
                                86400 )        ;minimum

```

## NS (Name Server)

```

<domain>    [<ttl>] [<class>]  NS    <server>

```

The NS record 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. If machines SRI-NIC.ARPA and C.ISI.EDU provide name lookup service for the domain COM then the following entries would be used:

```

      COM.      NS      SRI-NIC.ARPA.
                NS      C.ISI.EDU.

```

Note that the machines providing name service do not have to live in the named domain. There should be one NS record for each server for a domain. Also note that the name "COM" defaults for the second NS record.

NS records for a domain exist in both the zone that delegates the domain, and in the domain itself.

## GLUE RECORDS

If the name server host for a particular domain is itself inside the domain, then a 'glue' record will be needed. A glue record is an A (address) RR that specifies the address of the server. Glue records are only needed in the server delegating the domain, not in the domain itself. If for example the name server for domain SRI.COM was KL.SRI.COM, then the NS record would look like this, but you will also need to have the following A record.

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

## A (Address)

```
<host>    [<ttl>] [<class>]  A    <address>
```

The data for an A record is an internet address in dotted decimal form. A sample A record might look like:

```
SRI-NIC.ARPA.          A          10.0.0.51
```

There should be one A record for each address of a host.

## CNAME ( Canonical Name)

```
<nickname>    [<ttl>] [<class>]  CNAME    <host>
```

The CNAME record is used for nicknames. The name associated with the RR is the nickname. The data portion is the official name. For example, a machine named SRI-NIC.ARPA may want to have the nickname NIC.ARPA. In that case, the following RR would be used:

```
NIC.ARPA.          CNAME    SRI-NIC.ARPA.
```

There must not be any other RRs associated with a nickname of the same class.

Nicknames are also useful when a host changes it's name. In that case, it is usually a good idea to have a CNAME pointer so that people still using the old name will get to the right place.

## HINFO (Host Info)

```
<host>    [<ttl>] [<class>]    HINFO    <hardware>    <software>
```

The HINFO record gives information about a particular host. The data is two strings separated by whitespace. The first string is a hardware description and the second is software. The hardware is usually a manufacturer name followed by a dash and model designation. The software string is usually the name of the operating system.

Official HINFO types can be found in the latest Assigned Numbers RFC, the latest of which is RFC-1010. The Hardware type is called the Machine name and the Software type is called the System name.

Some sample HINFO records:

```
SRI-NIC.ARPA.          HINFO    DEC-2060 TOPS20
UCBARPA.Berkeley.EDU.  HINFO    VAX-11/780 UNIX
```

## WKS (Well Known Services)

```
<host> [<ttl>] [<class>] WKS <address> <protocol> <services>
```

The WKS record is used to list Well Known Services a host provides. WKS's are defined to be services on port numbers below 256. The WKS record lists what services are available at a certain address using a certain protocol. The common protocols are TCP or UDP. A sample WKS record for a host offering the same services on all address would look like:

Official protocol names can be found in the latest Assigned Numbers RFC, the latest of which is RFC-1010.

```
SRI-NIC.ARPA.    WKS  10.0.0.51  TCP  TELNET FTP SMTP
                  WKS  10.0.0.51  UDP  TIME
                  WKS  26.0.0.73  TCP  TELNET FTP SMTP
                  WKS  26.0.0.73  UDP  TIME
```

## MX (Mail Exchanger) (See RFC-974 for more details.)

```
<name>    [<ttl>] [<class>]    MX    <preference>    <host>
```

MX records specify where mail for a domain name should be delivered. There may be multiple MX records for a particular name. The preference value specifies the order a mailer should try multiple MX records when delivering mail. Zero is the highest preference. Multiple records for the same name may have the same preference.



A host BAR.FOO.COM may want its mail to be delivered to the host PO.FOO.COM and would then use the MX record:

BAR.FOO.COM.	MX	10	PO.FOO.COM.
--------------	----	----	-------------

A host BAZ.FOO.COM may want its mail to be delivered to one of three different machines, in the following order:

BAZ.FOO.COM.	MX	10	PO1.FOO.COM.
	MX	20	PO2.FOO.COM.
	MX	30	PO3.FOO.COM.

An entire domain of hosts not connected to the Internet may want their mail to go through a mail gateway that knows how to deliver mail to them. If they would like mail addressed to any host in the domain FOO.COM to go through the mail gateway they might use:

FOO.COM.	MX	10	RELAY.CS.NET.
*.FOO.COM.	MX	20	RELAY.CS.NET.

Note that you can specify a wildcard in the MX record to match on anything in FOO.COM, but that it won't match a plain FOO.COM.

#### IN-ADDR.ARPA

The structure of names in the domain system is set up in a hierarchical way such that the address of a name can be found by tracing down the domain tree contacting a server for each label of the name. Because of this 'indexing' based on name, there is no easy way to translate a host address back into its host name.

In order to do the reverse translation easily, a domain was created that uses hosts' addresses as part of a name that then points to the data for that host. In this way, there is now an 'index' to hosts' RRs based on their address. This address mapping domain is called IN-ADDR.ARPA. Within that domain are subdomains for each network, based on network number. Also, for consistency and natural groupings, the 4 octets of a host number are reversed.

For example, the ARPANET is net 10. That means there is a domain called 10.IN-ADDR.ARPA. Within this domain there is a PTR RR at 51.0.0.10.IN-ADDR that points to the RRs for the host SRI-NIC.ARPA (who's address is 10.0.0.51). Since the NIC is also on the MILNET (Net 26, address 26.0.0.73), there is also a PTR RR at 73.0.0.26.IN-ADDR.ARPA that points to the same RR's for SRI-NIC.ARPA. The format of these special pointers is defined below along with the examples for the NIC.

## PTR

```
<special-name>    [<ttl>] [<class>]    PTR    <name>
```

The PTR record is used to let special names point to some other location in the domain tree. They are mainly used in the IN-ADDR.ARPA records for translation of addresses to names. PTR's should use official names and not aliases.

For example, host SRI-NIC.ARPA with addresses 10.0.0.51 and 26.0.0.73 would have the following records in the respective zone files for net 10 and net 26:

```
51.0.0.10.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
73.0.0.26.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
```

## GATEWAY PTR's

The IN-ADDR tree is also used to locate gateways on a particular network. Gateways have the same kind of PTR RRs as hosts (as above) but in addition they have other PTRs used to locate them by network number alone. These records have only 1, 2, or 3 octets as part of the name depending on whether they are class A, B, or C networks, respectively.

Lets take the SRI-CSL gateway for example. It connects 3 different networks, one class A, one class B and one class C. It will have the standard RR's for a host in the CSL.SRI.COM zone:

```
GW.CSL.SRI.COM.      A      10.2.0.2
                     A      128.18.1.1
                     A      192.12.33.2
```

Also, in 3 different zones (one for each network), it will have one of the following number to name translation pointers:

```
2.0.2.10.IN-ADDR.ARPA.    PTR    GW.CSL.SRI.COM.
1.1.18.128.IN-ADDR.ARPA.  PTR    GW.CSL.SRI.COM.
1.33.12.192.IN-ADDR.ARPA. PTR    GW.CSL.SRI.COM.
```

In addition, in each of the same 3 zones will be one of the following gateway location pointers:

```
10.IN-ADDR.ARPA.          PTR    GW.CSL.SRI.COM.
18.128.IN-ADDR.ARPA.      PTR    GW.CSL.SRI.COM.
33.12.192.IN-ADDR.ARPA.   PTR    GW.CSL.SRI.COM.
```

## INSTRUCTIONS

Adding a subdomain.

To add a new subdomain to your domain:

Setup the other domain server and/or the new zone file.

Add an NS record for each server of the new domain to the zone file of the parent domain.

Add any necessary glue RRs.

Adding a host.

To add a new host to your zone files:

Edit the appropriate zone file for the domain the host is in.

Add an entry for each address of the host.

Optionally add CNAME, HINFO, WKS, and MX records.

Add the reverse IN-ADDR entry for each host address in the appropriate zone files for each network the host is on.

Deleting a host.

To delete a host from the zone files:

Remove all the hosts' resource records from the zone file of the domain the host is in.

Remove all the hosts' PTR records from the IN-ADDR zone files for each network the host was on.

Adding gateways.

Follow instructions for adding a host.

Add the gateway location PTR records for each network the gateway is on.

Deleting gateways.

Follow instructions for deleting a host.

Also delete the gateway location PTR records for each network

the gateway was on.

#### COMPLAINTS

These are the suggested steps you should take if you are having problems that you believe are caused by someone else's name server:

1. Complain privately to the responsible person for the domain. You can find their mailing address in the SOA record for the domain.
2. Complain publicly to the responsible person for the domain.
3. Ask the NIC for the administrative person responsible for the domain. Complain. You can also find domain contacts on the NIC in the file NETINFO:DOMAIN-CONTACTS.TXT
4. Complain to the parent domain authorities.
5. Ask the parent authorities to excommunicate the domain.

## EXAMPLE DOMAIN SERVER DATABASE FILES

The following examples show how zone files are set up for a typical organization. SRI will be used as the example organization. SRI has decided to divided their domain SRI.COM into a few subdomains, one for each group that wants one. The subdomains are CSL and ISTC.

Note the following interesting items:

There are both hosts and domains under SRI.COM.

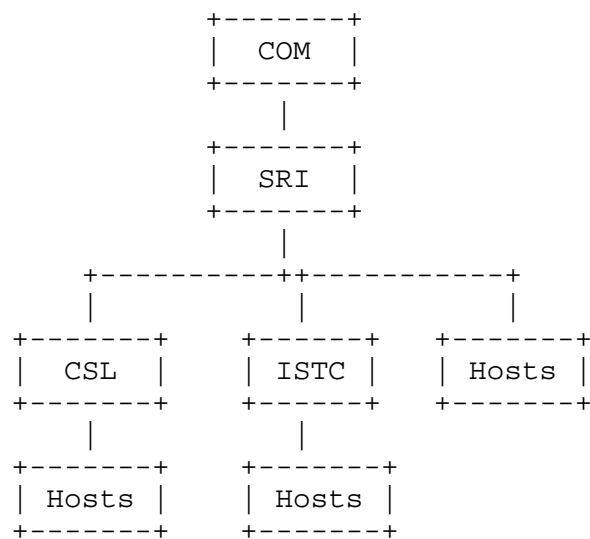
CSL.SRI.COM is both a domain name and a host name.

All the domains are serviced by the same pair of domain servers.

All hosts at SRI are on net 128.18 except hosts in the CSL domain which are on net 192.12.33. Note that a domain does not have to correspond to a physical network.

The examples do not necessarily correspond to actual data in use by the SRI domain.

## SRI Domain Organization



[File "CONFIG.CMD". Since bootstrap files are not standardized, this file is presented using a pseudo configuration file syntax.]

```
load root server list           from file ROOT.SERVERS
load zone SRI.COM.              from file SRI.ZONE
load zone CSL.SRI.COM.         from file CSL.ZONE
load zone ISTD.SRI.COM.        from file ISTD.ZONE
load zone 18.128.IN-ADDR.ARPA.  from file SRINET.ZONE
load zone 33.12.192.IN-ADDR.ARPA. from file SRI-CSL-NET.ZONE
```

[File "ROOT.SERVERS". Again, the format of this file is not standardized.]

;list of possible root servers

SRI-NIC.ARPA	10.0.0.51	26.0.0.73	
C.ISI.EDU	10.0.0.52		
BRL-AOS.ARPA	192.5.25.82	192.5.22.82	128.20.1.2
A.ISI.EDU	26.3.0.103		

[File "SRI.ZONE"]

```
SRI.COM.      IN      SOA      KL.SRI.COM. DLE.STRIPE.SRI.COM. (
                                870407  ;serial
                                1800    ;refresh every 30 minutes
                                600     ;retry every 10 minutes
                                604800  ;expire after a week
                                86400   ;default of an hour
                                )
```

```
SRI.COM.      NS      KL.SRI.COM.
               NS      STRIPE.SRI.COM.
               MX      10      KL.SRI.COM.
```

;SRI.COM hosts

```
KL            A      10.1.0.2
               A      128.18.10.6
               MX     10      KL.SRI.COM.
```

```
STRIPE        A      10.4.0.2
STRIPE        A      128.18.10.4
               MX     10      STRIPE.SRI.COM.
```

```
NIC           CNAME   SRI-NIC.ARPA.
```

```
Blackjack      A      128.18.2.1
               HINFO   VAX-11/780      UNIX
               WKS     128.18.2.1      TCP TELNET FTP
```

```
CSL           A      192.12.33.2
               HINFO   FOONLY-F4      TOPS20
               WKS     192.12.33.2    TCP TELNET FTP SMTP FINGER
               MX     10      CSL.SRI.COM.
```



[File "CSL.ZONE"]

```

CSL.SRI.COM.      IN      SOA      KL.SRI.COM. DLE.STRIPE.SRI.COM. (
                                870330 ;serial
                                1800   ;refresh every 30 minutes
                                600    ;retry every 10 minutes
                                604800 ;expire after a week
                                86400  ;default of a day
                                )

CSL.SRI.COM.      NS
                  NS      KL.SRI.COM.
                  A       STRIPE.SRI.COM.
                        192.12.33.2

```

;CSL.SRI.COM hosts

```

A      CNAME      CSL.SRI.COM.
B      A          192.12.33.3
      HINFO      FOONLY-F4      TOPS20
      WKS        192.12.33.3      TCP TELNET FTP SMTP
GW     A          10.2.0.2
      A          192.12.33.1
      A          128.18.1.1
      HINFO      PDP-11/23      MOS
SMELLY A          192.12.33.4
      HINFO      IMAGEN        IMAGEN
SQUIRREL A        192.12.33.5
      HINFO      XEROX-1100      INTERLISP
VENUS  A          192.12.33.7
      HINFO      SYMBOLICS-3600  LISPM
HELIUM A          192.12.33.30
      HINFO      SUN-3/160      UNIX
ARGON  A          192.12.33.31
      HINFO      SUN-3/75      UNIX
RADON  A          192.12.33.32
      HINFO      SUN-3/75      UNIX

```

[File "ISTC.ZONE"]

```
ISTC.SRI.COM.    IN    SOA      KL.SRI.COM. roemers.JOYCE.ISTC.SRI.COM. (
                        870406      ;serial
                        1800      ;refresh every 30 minutes
                        600       ;retry every 10 minutes
                        604800    ;expire after a week
                        86400     ;default of a day
                        )
```

```
ISTC.SRI.COM.    NS          KL.SRI.COM.
                  NS          STRIPE.SRI.COM.
                  MX          10      SPAM.ISTC.SRI.COM.
```

; ISTC hosts

```
joyce      A      128.18.4.2
            HINFO  VAX-11/750 UNIX
bozo       A      128.18.0.6
            HINFO  SUN UNIX
sundae     A      128.18.0.11
            HINFO  SUN UNIX
tsca       A      128.18.0.201
            A      10.3.0.2
            HINFO  VAX-11/750 UNIX
            MX     10  TSCA.ISTC.SRI.COM.
tsc        CNAME  tsca
prmh       A      128.18.0.203
            A      10.2.0.51
            HINFO  PDP-11/44 UNIX
spam       A      128.18.4.3
            A      10.2.0.107
            HINFO  VAX-11/780 UNIX
            MX     10  SPAM.ISTC.SRI.COM.
```

[File "SRINET.ZONE"]

```
18.128.IN-ADDR.ARPA.    IN  SOA  KL.SRI.COM  DLE.STRIPE.SRI.COM. (
                        870406  ;serial
                        1800    ;refresh every 30 minutes
                        600     ;retry every 10 minutes
                        604800  ;expire after a week
                        86400   ;default of a day
                        )
```

```
18.128.IN-ADDR.ARPA.    NS      KL.SRI.COM.
                        NS      STRIPE.SRI.COM.
                        PTR     GW.CSL.SRI.COM.
```

; SRINET [128.18.0.0] Address Translations

; SRI.COM Hosts

```
1.2.18.128.IN-ADDR.ARPA.    PTR      Blackjack.SRI.COM.
```

; ISTD.SRI.COM Hosts

```
2.4.18.128.IN-ADDR.ARPA.    PTR      joyce.ISTD.SRI.COM.
6.0.18.128.IN-ADDR.ARPA.    PTR      bozo.ISTD.SRI.COM.
11.0.18.128.IN-ADDR.ARPA.   PTR      sundae.ISTD.SRI.COM.
201.0.18.128.IN-ADDR.ARPA.  PTR      tsca.ISTD.SRI.COM.
203.0.18.128.IN-ADDR.ARPA.  PTR      prmh.ISTD.SRI.COM.
3.4.18.128.IN-ADDR.ARPA.    PTR      spam.ISTD.SRI.COM.
```

; CSL.SRI.COM Hosts

```
1.1.18.128.IN-ADDR.ARPA.    PTR      GW.CSL.SRI.COM.
```

[File "SRI-CSL-NET.ZONE"]

```
33.12.192.IN-ADDR.ARPA. IN SOA KL.SRI.COM DLE.STRIPE.SRI.COM. (  
    870404 ;serial  
    1800 ;refresh every 30 minutes  
    600 ;retry every 10 minutes  
    604800 ;expire after a week  
    86400 ;default of a day  
)
```

```
33.12.192.IN-ADDR.ARPA. NS      KL.SRI.COM.  
                        NS      STRIPE.SRI.COM.  
                        PTR      GW.CSL.SRI.COM.
```

; SRI-CSL-NET [192.12.33.0] Address Translations

; SRI.COM Hosts

```
2.33.12.192.IN-ADDR.ARPA.      PTR      CSL.SRI.COM.
```

; CSL.SRI.COM Hosts

```
1.33.12.192.IN-ADDR.ARPA.      PTR      GW.CSL.SRI.COM.  
3.33.12.192.IN-ADDR.ARPA.      PTR      B.CSL.SRI.COM.  
4.33.12.192.IN-ADDR.ARPA.      PTR      SMELLY.CSL.SRI.COM.  
5.33.12.192.IN-ADDR.ARPA.      PTR      SQUIRREL.CSL.SRI.COM.  
7.33.12.192.IN-ADDR.ARPA.      PTR      VENUS.CSL.SRI.COM.  
30.33.12.192.IN-ADDR.ARPA.     PTR      HELIUM.CSL.SRI.COM.  
31.33.12.192.IN-ADDR.ARPA.     PTR      ARGON.CSL.SRI.COM.  
32.33.12.192.IN-ADDR.ARPA.     PTR      RADON.CSL.SRI.COM.
```

## APPENDIX

BIND (Berkeley Internet Name Domain server) distributed with 4.3 BSD UNIX

This section describes two BIND implementation specific files; the boot file and the cache file. BIND has other options, files, and specifications that are not described here. See the Name Server Operations Guide for BIND for details.

The boot file for BIND is usually called "named.boot". This corresponds to file "CONFIG.CMD" in the example section.

```

-----
cache          .                named.ca
primary        SRI.COM          SRI.ZONE
primary        CSL.SRI.COM      CSL.ZONE
primary        ISTC.SRI.COM     ISTC.ZONE
primary        18.128.IN-ADDR.ARPA SRINET.ZONE
primary        33.12.192.IN-ADDR.ARPA SRI-CSL-NET.ZONE
-----

```

The cache file for BIND is usually called "named.ca". This corresponds to file "ROOT.SERVERS" in the example section.

```

-----
;list of possible root servers
.      1      IN    NS    SRI-NIC.ARPA.
                        NS    C.ISI.EDU.
                        NS    BRL-AOS.ARPA.
                        NS    C.ISI.EDU.

;and their addresses
SRI-NIC.ARPA.          A    10.0.0.51
                        A    26.0.0.73
C.ISI.EDU.             A    10.0.0.52
BRL-AOS.ARPA.          A    192.5.25.82
                        A    192.5.22.82
                        A    128.20.1.2
A.ISI.EDU.             A    26.3.0.103
-----

```

## REFERENCES

- [1] Dunlap, K., "Name Server Operations Guide for BIND", CSRG, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California.
- [2] Partridge, C., "Mail Routing and the Domain System", RFC-974, CSNET CIC BBN Laboratories, January 1986.
- [3] Mockapetris, P., "Domains Names - Concepts and Facilities", RFC-1034, USC/Information Sciences Institute, November 1987.
- [4] Mockapetris, P., "Domain Names - Implementations Specification", RFC-1035, USC/Information Sciences Institute, November 1987.

