Previous Page | Next Page

**Domain Name Service**

## Host Names

Domain Name Service (DNS) is the service used to convert human readable names of hosts to IP addresses. Host names are not case sensitive and can contain alphabetic or numeric letters or the hyphen. Avoid the underscore. A fully qualified domain name (FQDN) consists of the host name plus domain name as in the following example: computername.domain.com

The part of the system sending the

queries is called the resolver and is the client side of the configuration. The nameserver answers the queries. Read RFCs 1034 and 1035. These contain the bulk of the DNS information and are superceded by RFCs 1535-1537. Naming is in RFC 1591. The main function of DNS is the mapping of IP addresses to human readable names.

Three main components of DNS

1. resolver
2. name server
3. database of resource records(RRs)

## Domain Name System

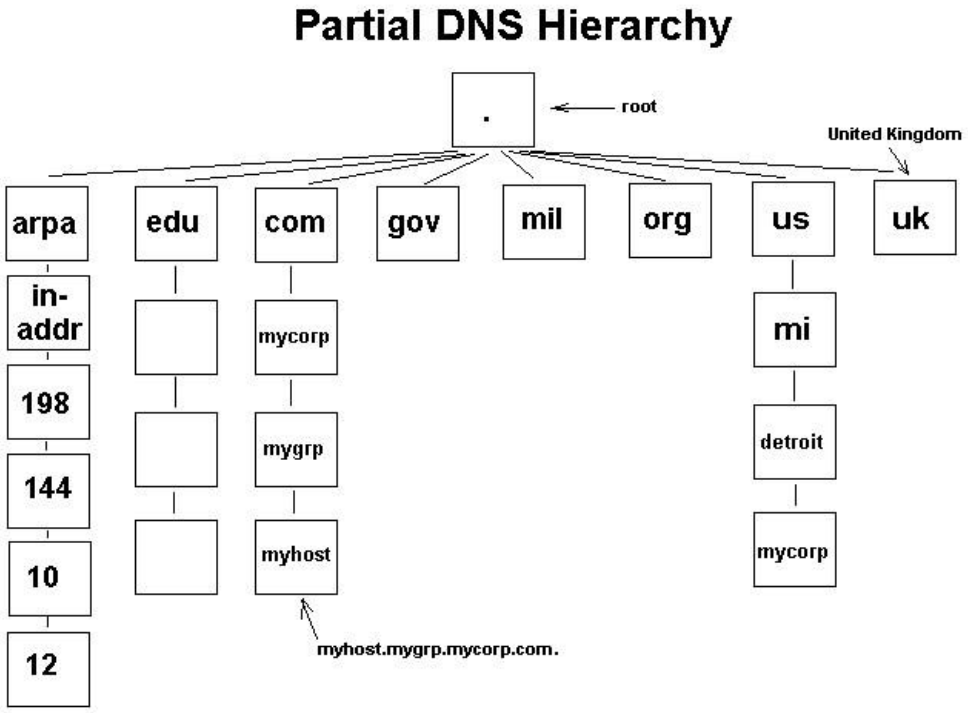
The Domain Name System (DNS) is basically a large database which resides on various computers and it contains the names and IP addresses of various hosts on the internet and various domains. The Domain Name System is used to provide information to the Domain Name Service to use when queries are made. The service is the act of querying the database, and the system is the data structure and data itself. The Domain Name System is similar to a file system in Unix or DOS starting with a root. Branches attach to the root to create a huge set of paths. Each branch in the DNS is called a label. Each label can be 63 characters long, but most are less. Each text word between the dots can be 63 characters in length, with the total domain name (all the labels) limited to 255 bytes in overall length. The domain name system database is divided into sections called **zones**. The name servers in their respective zones are responsible for answering queries for their zones. A zone is a subtree of DNS and is administered separately. There are multiple name servers for a zone. There is usually one primary nameserver and one or more secondary name servers. A name server may be authoritative for more than one zone.

DNS names are assigned through the Internet Registries by the Internet Assigned Number Authority (IANA). The domain name is a name assigned to an internet domain. For example, mycollege.edu represents the domain name of an educational institution. The names microsoft.com and 3Com.com represent the domain names at those commercial companies. Naming hosts within the domain is up to individuals administer their domain.

Access to the Domain name database is through a resolver which may be a program or part of an operating system that resides on users workstations. In Unix the resolver is accessed by using the library functions "gethostbyname" and "gethostbyaddr". The resolver will send requests to the name servers to return information requested by the user. The requesting computer tries to connect to the name server using its IP address rather than the name.

## Structure and message format

The drawing below shows a partial DNS hierarchy. At the top is what is called the root and it is the start of all other branches in the DNS tree. It is designated with a period. Each branch moves down from level to level. When referring to DNS addresses, they are referred to from the bottom up with the root designator (period) at the far right. Example: "myhost.mycompany.com.".




DNS is hierarchical in structure. A domain is a subtree of the domain name space. From the root, the assigned toplevel domains in the U.S. are:

- GOV - Government body.
- EDU - Educational body.
- INT - International organization
- NET - Networks COM - Commercial entity.
- MIL - U. S. Military.

ORG - Any other organization not previously listed.

Outside this list are top level domains for various countries.

Each node on the domain name system is separated by a ".". Example: "mymachine.mycompany.com.". Note that any name ending in a "." is an absolute domain name since it goes back to root.

## DNS Message format:

| Bits | Name | Description |
|---|---|---|
| 0-15 | Identification | Used to match responses to requests. Set by client and returned by server. |
| 16-31 | Flags | Tells if query or response, type of query, if authoritative answer, if truncated, if recursion desired, and if recursion is available. |
| 32-47 | Number of uestions | |
| 48-63 | Number of answer RRs | |
| 64-79 | Number of authority RRs | |

| | | |
|---|---|---|
| 80-95 | Number of additional RRs | |
| 96-?? | Questions - variable lengths | There can be variable numbers of questions sent. |
| ??-?? | Answers - variable lengths | Answers are variable numbers of resource records. |
| ??-?? | Authority - variable lengths | |
| ??-?? | Additional Information variable lengths | |

Question format includes query name, query type and query class. The query name is the name being looked up. The query class is normally 1 for internet address. The query types are listed in the table below. They include NS, CNAME, A, etc.

The answers, authority and additional information are in resource record (RR) format which contains the following.

1. Domain name
2. Type - One of the RR codes listed below.
3. Class - Normally indicates internet data which is a 1.
4. Time to live field - The number of seconds the RR is saved by the client.
5. Resource data length specifies the amount of data. The data is dependent on its type such as CNAME, A, NS or others as shown in the table below. If the type is "A" the data is a 4 byte IP address.

**The table below shows resource record types:**

| Type | RR value | Description |
|---|---|---|
| A | 1 | Host's IP address |
| NS | 2 | Host's or domain's name server(s) |
| CNAME | 5 | Host's canonical name, host identified by an alias domain name |
| PTR | 12 | Host's domain name, host identified by its IP address |
| HINFO | 13 | Host information |
| MX | 15 | Host's or domain's mail exchanger |
| AXFR | 252 | Request for zone transfer |
| ANY | 255 | Request for all records |

## Usage and file formats

If a domain name is not found when a query is made, the server may search for the name elsewhere and return the information to the requesting workstation, or return the address of a name server that the workstation can query to get more information. There are special servers on the Internet that provide guidance to all name servers. These are known as root name servers. They do not contain all information about every host on the Internet, but they do provide direction as to where domains are located (the IP address of the name server for the uppermost domain a server is requesting). The root name server is the starting point to find any domain on the Internet.

## Name Server Types

There are three types of name servers:

1. The primary master builds its database from files that were preconfigured on its hosts, called zone or database files. The name server reads these files and builds a database for the zone it is authoritative for.
2. Secondary masters can provide information to resolvers just like the primary masters, but they get their information from the primary. Any updates to the database are provided by the primary.
3. Caching name server - It gets all its answers to queries from other name servers and saves (caches) the answers. It is a non-authoritative server.

The caching only name server generates no zone transfer traffic. A DNS Server that can communicate outside of the private network to resolve a DNS name query is referred to as **forwarder**.

### DNS Query Types

There are two types of queries issued:

1. **Recursive** queries received by a server forces that server to find the information requested or post a message back to the querier that the information cannot be found.
2. **Iterative** queries allow the server to search for the information and pass back the best information it knows about. This is the type that is used between servers. Clients used the recursive query.
3. **Reverse** - The client provides the IP address and asks for the name. In other queries the name is provided, and the IP address is returned to the client. Reverse lookup entries for a network 192.168.100.0 is "100.168.192.in-addr arpa".

Generally (but not always), a server-to-server query is iterative and a client-resolver-to-server query is recursive. You should also note that a server can be queried or it can be the person placing a query. Therefore, a server contains both the server and client functions. A server can transmit either type of query. If it is handed a recursive query from a remote source, it must transmit other queries to find the specified name, or send a message back to the originator of the query that the name could not be found.

### DNS Transport protocol

DNS resolvers first attempt to use UDP for transport, then use TCP if UDP fails.

### The DNS Database

A database is made up of records and the DNS is a database. Therefore, common resource record types in the DNS database are:

- A - Host's IP address. Address record allowing a computer name to be translated into an IP address. Each computer must have this record for its IP address to be located. These names are not assigned
- for clients that have dynamically assigned IP addresses, but are a must for locating servers with
- static IP addresses.
- PTR - Host’s domain name, host identified by its IP address

CNAME - Host’s canonical name allows additional names or aliases to be used to locate a computer.
MX - Host’s or domain’s mail exchanger.

- NS - Host’s or domain’s name server(s).
- SOA - Indicates authority for the domain
- TXT - Generic text record
- SRV - Service location record
- RP - Responsible person

HINFO - Host information record with CPU type and operating system.

When a resolver requests information from the server, the DNS query message indicates one of the preceding types.

### DNS Files

- CACHE.DNS - The DNS Cache file. **This file is used to resolve internet DNS queries**. On Windows systems, it is located in the WINNTROOT\system32\DNS directory and is used to configure a DNS server to use a DNS server on the internet to resolve names not in the local domain.

### Example Files

Below is a partial explanation of some records in the database on a Linux based system. The reader should view this information because it explains some important DNS settings that are common to all DNS servers. An example /var/named/db.mycompany.com.hosts file is listed below.

```
mycompany.com.          IN      SOA     mymachine.mycompany.com.        root.mymachine.mycompany.com.
                1999112701              ; Serial number as date and two digit number YYMMDDXX
                10800                   ; Refresh in seconds 28800=8H
                3600                    ; Retry in seconds 7200=2H
                604800                  ; Expire 3600000=1 week
                86400 )                 ; Minimum TTL 86400=24Hours
mycompany.com.                  IN        NS      mymachine.mycompany.com.
mycompany.com.                  IN        MX              10              mailmachine.mycompany.com.
mymachine.mycompany.com.        IN        A       10.1.0.100
mailmachine.mycompany.com.      IN        A       10.1.0.4
george.mycompany.com.           IN        A       10.1.3.16
```

A Line by line description is as follows:

1. The entries on this line are:
    1. mycompany.com. - Indicates this server is for the domain mycompany.com.
    2. IN - Indicates Internet Name.
    3. SOA - Indicates this server is the authority for its domain, mycompany.com.
    4. mymachine.mycompany.com. - The primary nameserver for this domain.
    5. root.mymachine.mycompany.com. - The person to contact for more information.

    The lines in the parenthesis, listed below, are for the secondary nameserver(s) which run as slave(s) to this one (since it is the master).
2. 1999112701 - Serial number - If less than master's SN, the slave will get a new copy of this file from the master.
3. 10800 - Refresh - The time in seconds between when the slave compares this file's SN with the master.
4. 3600 - Retry - The time the server should wait before asking again if the master fails to respond to a file update (SOA request).
5. 604800 - Expire - Time in seconds the slave server can respond even though it cannot get an updated zone file.
6. 86400 - TTL - The time to live (TTL) in seconds that a resolver will use data received from a nameserver before it will ask for the same data again.
7. This line is the nameserver resource record. There may be several of these if there are slave name servers.

```
mycompany.com.                  IN        NS      mymachine.mycompany.com.
```

   Add any slave server entries below this like:

```
mycompany.com.                  IN        NS      ournamesv1.mycompany.com.
mycompany.com.                  IN        NS      ournamesv2.mycompany.com.
mycompany.com.                  IN        NS      ournamesv3.mycompany.com.
```

8. This line indicates the mailserver record.

```
mycompany.com.                  IN        MX            10        mailmachine.mycompany.com
```

   There can be several mailservers. The numeric value on the line indicates the preference or precedence for the use of that mail server. A lower number indicates a higher preference. The range of values is from 0 to 65535. To enter more mailservers, enter a new line for each one similar to the nameserver entries above, but be sure to set the preferences value correctly, at different values for each mailserver.

9. The rest of the lines are the name to IP mappings for the machines in the organization. Note that the nameserver and mailserver are listed here with IP addresses along with any other server machines required for your network.

```
mymachine.mycompany.com.        IN A 10.1.0.100
mailmachine.mycompany.com.      IN A 10.1.0.4
george.mycompany.com.           IN A 10.1.3.16
```

Domain names written with a dot on the end are absolute names which specify a domain name exactly as it exists in the DNS hierarchy from the root. Names not ending with a dot may be a subdomain to some other domain.

Aliases are specified in lines like the following:

```
mymachine.mycompany.com      IN  CNAME  nameserver.mycompany.com.
george.mycompany.com  IN     CNAME  dataserver.mycompany.com.
Linux1.mycompany.com  IN     CNAME  engserver.mycompany.com.
Linux2.mycompany.com  IN     CNAME  mailserver.mycompany.com.
```

When a client (resolver) sends a request, if the nameserver finds a CNAME record, it replaces the requested name with the CNAME, then finds the address of the CNAME value, and return this value to the client.

A host that has more than one network card which is set to address two different subnets can have more than one address for a name.

```
  mymachine.mycompany.com      IN     A     10.1.0.100
                               IN     A     10.1.1.100
```

When a client queries the nameserver for the address of a multi homed host, the nameserver will return the address that is closest to the client address. If the client is on a different network than both the subnet addresses of the multi homed host, the server will return both addresses.