

Table Of Contents

Table Of Contents 1

Resolving DNS names to IP addresses 2

Further readings: 3

[Home](#) > [Faq](#) > [Bind dns](#)

DNS Name Resolution

Posted by [Vivek Gite](#) <vivek@nixcraft.com>

Q. Can you explain DNS name resolution concept? Why I need to set DNS resolution for my Windows or Linux server? How do I setup DNS name resolution under Linux / UNIX server system?



A. DNS name resolution is nothing but resolving host names, such as www.nixcraft.com, to their corresponding IP addresses. DNS works as the "phone book" for the Internet by translating hostname into IP address or vice versa. Most DNS server stores following information:

[1]

- a) **Hostname** and their **IP address**
- b) List of **mail server and their IP address** for given domain name
- c) **Anti spam configuration** and much more.

Without DNS name resolution, nothing will work on the Internet. Nobody likes to remember IP address, so DNS is foundation of many Internet services such as web, proxy, email and so on.

Resolving DNS names to IP addresses

When you type www.yahoo.com into a web browser, the application has to find out IP address associated with www.yahoo.com. Each part of network has DNS server or name servers. Each application send a request called dns lookup to DNS server. Each DNS server has limited information about host names and ip address. Almost all DNS server constantly query each other to get information using root servers.

Each computer is configured to query specific name server. Usually home computers are configure to query ISP name servers or [free dns name servers](#) ^[2]. Here is a typical UNIX / Linux /etc/resolv.conf file with nameserver IP address:

```
$ cat /etc/resolv.conf
```

Sample output:

```
nameserver 208.67.222.222
nameserver 208.67.220.220
```

Each application can find www.yahoo.com IP address by sending a request to 208.67.222.222 or 208.67.220.220 IP address. This procedure is called **hostname resolution** and the algorithm that performs this operation is called the **resolver**. Let us see how to find out IP address for freebsd.nixcraft.in hostname:

1. The web browser will check local cache database to find out answer. If it can get an answer directly from these, it proceeds no further.
2. Otherwise request will be sent to nameserver IP 208.67.222.222 to find IP address for freebsd.nixcraft.in host.
3. 208.67.222.222 server will decide if that IP has been recently looked up before. If it has, there is no need to ask further, since the result would be stored in a local cache.
4. 208.67.222.222 will see if the domain is local. I.e. if it is a computer that it has direct information about. In this case this would only be true if the 208.67.222.222 were Obsidian's very own name server.
5. 208.67.222.222 will strip out the TLD (Top Level Domain) .in It will query a root name server, asking what name server is responsible for .IN. Depend upon the answer 208.67.222.222 will query authoritative server for IP address.
6. 208.67.222.222 will return the result to the application.
7. 208.67.222.222 will store each of these results in a local cache with an expiry date. To avoid having to look them up a second time.

Please note that end users generally do not communicate directly with a DNS resolver. Instead DNS-resolution takes place transparently in client-applications such as web-browsers, mail-clients, and other Internet applications. When an application makes a request which requires a DNS lookup, such programs send a resolution request to the local DNS resolver in the local operating system, which in turn handles the communications required. All you have to do is configure your system for dns name resolution.

Further readings:

DNS is complicated topic I recommend following resources:

- [DNS and BIND \(5th Edition\)](#) ^[3] - This is the definitive book on the Domain Name System (DNS). This book's early chapters give a view of DNS from high altitude, explaining basic concepts such as domains, name servers, and name resolution. From there, the authors proceed on a more practical tack, presenting specific instructions for setting up your own domain and DNS server using BIND. The authors then tell you what to do as your domain grows and you need to add more machines, subdomains, and greater throughput capacity. They also talk a lot about nslookup and C programming with the various DNS and BIND libraries. Administrators will find the chapter on BIND debugging output particularly helpful. Here, the authors translate BIND's mysterious error messages and offer specific strategies for fixing and optimizing the program.
- [DNS and Name Resolution](#) ^[4]
- man pages resolver, named, resolv.conf

4000+ howtos and counting! Want to read more Linux / UNIX howtos, tips and tricks? Subscribe to our [daily email](#) newsletter or [weekly newsletter](#) to make sure you don't miss a single tip/tricks. Alternatively, subscribe via [RSS/XML](#) feed.

Article printed from Frequently Asked Questions About Linux / UNIX: <http://www.cyberciti.biz/faq/>

URL to article: <http://www.cyberciti.biz/faq/dns-name-resolution/>

URLs in this post:

[1] Image: <http://www.cyberciti.biz/faq/faq/category/unix/>

[2] free dns name servers: <http://www.cyberciti.biz/faq/free-dns-server/>

[3] DNS and BIND (5th Edition): <http://www.amazon.com/gp/redirect.html?ie=UTF8&location=http%3A%2F%2Fwww.amazon.com%2FDNS-BIND-5th-Cricket-Liu%2Fdp%2F0596100574%3Fie%3DUTF8%26s%3Dbooks%26qid%3D1213122348%26sr%3D8-1&tag=cyberciti-20&linkCode=ur2&camp=1789&creative=9325>

[4] DNS and Name Resolution: <http://burks.brighton.ac.uk/burks/linux/rute/node23.htm>