# Berkeley Internet Name Domain (BIND9)

The Domain Name System, or DNS, is one of the Internet's fundamental building blocks. It is the global, hierarchical, and distributed host information database that's responsible for translating names into addresses and vice versa.

## Examining DNS Record Types

Before we delve into creating a zone (an administrative domain) in DNS, we need to talk about record types. DNS provides not only host name–to-IP address translation, but also mail server information.

A record type tells DNS what kind of information it is storing. This record type is also something you can explicitly query. For example, you can query the MX record for a domain to find out what mail server to use when sending mail to a recipient in the domain. Services that rely on DNS (such as mail or name resolution) query a specific record in the DNS database for specific hosts. This drastically reduces the amount of traffic that is produced because you see only the data you explicitly asked for, and not the whole record. What makes this happen is the use of record types. With DNS, you have six main record types to keep in mind:

1. **Address record**—the most common record in DNS is the Address record. An Address record is used to translate a host name to an IP address.
2. **Pointer record**—a Pointer record is the reverse of an Address record. It translates an IP address to a host name.
3. **CNAME record**—If you want to create an alias of one host to another, a CNAME entry is used.
4. **MX record**—an MX record is used to define a Mail Exchanger for the domain (or zone) you have created.
5. **NS record**—the NS record is used to define the nameserver for this domain.
6. **SOA record**—the Start of Authority (SOA) is effectively the header for the zone in question. It contains information about the zone itself and is mandatory.

## Working with Zones

A zone is a description, or a definition, of a domain (or subdomain). The zone is what makes up the database full of records for a domain. Zone files are separate files located in **/etc/bind** and are text files containing the data for the zone. BIND defines its zones in **/etc/named.conf.default-zones** by specifying the location of these zone files. We will look at how a DNS zone is constructed and what information you can store for a specific zone.

We use **mynewsite.ie** as an example of a zone and will go through the DNS records and explain their uses.

**;**

**$TTL 604800**

**mynewsite.ie. IN SOA aoife.mynewsite.ie. john.mynewsite.ie. (**

**201103111 ;Serial**

**10800 ;Refresh**

**3600 ;Retry**

**604800 ;Expire**

**38400 ) ;Negative Cache TTL**

**NS aoife**

**MX 10 mail**

**MX 20 pirhana.ucd.ie.**

**mail A 212.13.208.115**

**aoife A 212.13.208.115**

**www CNAME aoife**

**niamh A 212.13.208.150**

## The Start of Authority

At the start of the zone, you have the SOA record. The Start of Authority dictates that this zone is authoritative for the domain in question, mynewsite.ie. Notice that mynewsite.ie ends in a full stop. This is extremely important in the zone file for any domain. A full stop is the delimiter for the end of the DNS tree, following the mynewsite.ie domain all the way up the tree, the full domain name is mynewsite.ie (with the full stop). If a full stop is not found, as in the aoife record listed at the end in the example, the SOA’s domain will be appended to the host name in the record.

## The SOA server

After the definition of the domain you are managing, you need to define the server that is authoritative for the domain. BIND is aware that it needs to find the IP address for the server from its zone file.

## The hostmaster

As with most things on the Internet, it is common practice to provide a technical contact for the service. In this case, it is the email address john.mynewsite.ie. You will notice that there is no @ sign in the email address, but a full stop. The hostmaster for the mynewsite.ie zone is john.mynewsite.ie (john@mynewsite.ie).

## The SOA record

The brackets around the rest of the data dictate that everything else up to the closing bracket is part of the SOA record. All time settings are in seconds.

## The serial number

The first entry is the serial number for the zone. This is one of the most important parts of the SOA because it must be changed any time you edit the zone file. It is the serial number that tells other DNS servers that are querying your DNS server that data has changed. If you do not change the serial number, your changes will not get propagated through the system. The general form of the serial number is the date, followed by an arbitrary number. For the 30th of November, you use 2010113001 (November 30th, 2010). Notice that the date is in reverse, with the year (2010), month (11), and day (30), with an additional two digits able to represent multiple changes in one day.

## The refresh rate

If you have a slave DNS server in your system (as a backup to your master), the refresh rate tells the slave server how often to check for updates to the zone. The refresh rate set to 10,800 seconds (3 hours).

## The retry rate

If your slave server cannot contact the master, the retry rate is how often it will attempt to contact the master. The retry rate to 3,600 seconds (1 hour).

## The expiry time

If the slave server cannot update the zone data in this time, it stops functioning. The expiry time to 604,800 seconds (1 week).

## The Time to Live

TTL (Time to Live, the amount of time before a name server that has requested a record considers it stale) of 604,800 seconds, or seven days.

## The NS entry

Every zone should have an NS entry (nameserver), and in this case, the nameserver for mynewsite.ie is the machine aoife (remember that with no full stop, it gets expanded to aoife.mynewsite.ie).

## The Mail Exchanger

If you wish to receive mail for your domain, you must specify an MX record for it. The MX record is used to define the host that receives mail for this domain. When an SMTP server needs to find the host that handles mail for a domain, it will query the MX record for the machine to connect to.

You can see a number in the second field of this record, and this is very important. The number is a preference order for the MX host specified. The lower the number, the higher the preference. So in this case, all SMTP transactions for this domain will attempt to connect to mail.mynewsite.ie, and if that fails they will try the machine pirhana.ucd.ie (the backup MX server).

The host machine for the domain is called mail.mynewsite.ie according to its IP entry in DNS, but we have an alias configured, calling the machine aoife. When defining an MX record, you must not use a CNAME record; it must be an Address record.

## The Address record

We have defined a machine called aoife (no full stop), with the IP address of 212.13.208.115. This is the record that points a host name to an IP address.

## The CNAME record

To create an alias of a host so that a lookup returns the same IP address, you use a CNAME record. It is an alias for a host name.

# Installing BIND

If you didn't install BIND during the initial Ubuntu install you can use **tasksel**. Alternatively you can just run **# apt-get install bind9 bind9-doc**

This default setup configures a caching name server for recursive queries.

## Caching Name Server

Essentially a caching name server acts as a middle- man for DNS queries. Once you configure some hosts to point to a caching name server, when one host requests a particular record, the caching name server goes out to the Internet, retrieves a record, and stores that record locally. If a second host requests the same record, and that record hasn’t expired from the cache, a caching name server will simply return the cached result. This can dramatically improve the response times for a network of hosts, especially for Web browsing, since multiple hosts request often the same records.

## BIND - file locations

* **/etc/bind/** This directory contains the main BIND configuration file, named.conf, as well as any individual zone files. Any new master zone files should also be stored in this directory or, if you have many zone files you wish to organize, in a subdirectory below **/etc/bind**.
* **/etc/bind/named.conf** This is BIND’s main configuration file and is where you change BIND’s options and behaviour. All of BIND’s individual zone files (files containing name and IP information) are also referenced here.
* **/etc/bind/db.\*** As a convention, all zone files start with db. and then some name or number to identify the particular zone. Names are typically used when the zone contains traditional forward DNS records (names mapped to IPs), and numbers are typically used for reverse DNS records (IPs mapped back to a name). For instance, by default Ubuntu’s BIND will include a few zone files such as db.root (information about the root name servers on the Internet), db.local (localhost zone information), and db.127 (reverse DNS records for localhost).
* **/var/cache/bind** This is BIND’s working directory and where it will store slave zone files. If your server will act as a slave for a particular zone, configure it to store its files here.
* **/etc/init.d/bind9** This is BIND’s init script. Once you install the bind9 package, it will automatically be set up to start on system boot, but you can run the init script manually with **/etc/init.d/bind9**, or **service bind9**.
* **/var/log/syslog** This is the default log file for BIND. A number of different services log to this file, but log entries for BIND will be prefixed by the key- word named, so if you wanted to see only the BIND log entries you could run **grep named /var/log/syslog**.

## DNS Master

Caching name servers are very useful, but when most people install BIND, they intend to host some zone files of their own. When a name server hosts zone files locally and doesn’t need to retrieve them from any other source, it is known as a master. When you want to add zones to a DNS server, there are basically three steps:

1. Create a zone file
2. Add a reference to that zone file in named.conf
3. Tell BIND to reload its configuration.

For my example let’s assume that I have a name server inside my network at 192.168.0.5 and I registered mynewsite.ie. I want this name server to have the following entries:

* ns1.mynewsite.ie points to192.168.0.5(the nameserver itself).
* [www.mynewsite.ie](http://www.example.net) points to192.168.0.7.
* gw.mynewsite.ie points to192.168.0.1.

The simplest way to create a new zone file is to copy one you already have and change it. In this case the best candidate is the **/etc/bind/db.local** file, so I copy it to **db.mynewsite.ie**:

**# cp /etc/bind/db.local /etc/bind/db.mynewsite.ie**

When I open db.mynewsite.ie in a text editor, I will see the following configuration:

;

; BIND data file for local loopback interface

;

$TTL 604800

@ IN SOA localhost. root.localhost. (

2010032101 ;Serial

604800 ;Refresh (7 days)

86400 ;Retry

2419200 ;Expire

604800 ) ;Negative Cache TTL

;

@ IN NS localhost.

@ IN A 127.0.0.1

@ IN AAAA ::1

The file lists localhost as a name server for this zone with an NS record, then sets localhost’s IP address to be 127.0.0.1 and even adds an IPv6 address for localhost with the AAAA record. I then changed this record to suit the requirements I set out previously, and this is the resulting zone file:

;

; BIND data file for local loopback interface

;

$TTL 1d

@ IN SOA ns1.mynewsite.ie. root.mynewsite.ie. (

2013102102 ;Serial

2d ;Refresh (7 days)

1d ;Retry

3d ;Expire

2h ) ;Negative Cache TTL

;

@ IN NS ns1.mynewsite.ie.

ns1 IN A 192.168.0.5

www IN A 192.168.0.7

gw IN A 192.168.0.1

I set ns1.mynewsite.ie as the SOA and root@mynewsite.ie as the contact e-mail address. I also set ns1.mynewsite.ie as the name server to use for this zone, but notice that the NS record references the name of the host. Since this name server is in the same domain, mynewsite.ie, I needed to make sure I added a record for it here that listed its IP address. Also note that I didn’t need to add .mynewsite.ie to any of the A records.

*Be sure once you save your zone file that it has the same permissions and ownership as the other zone files in the directory—that’s the best way to avoid any permission headaches later on once you reload BIND.*

Now that I have created the zone file, I need to add a reference to it in **named.conf.default-zones**. When I open it with my text editor, I can see a number of similar zone examples such as the one for the db.local file:

zone "localhost" { type master;

file "/etc/bind/db.local";

};

So I just add a similar entry to the very bottom of the file:

zone "mynewsite.ie" { type master;

file "/etc/bind/db.mynewsite.ie";

};

Once I save my changes, I just need to tell BIND to reload its configuration, and then I should be able to query the name server for one of the new records:

# service bind9 reload

\* Reloading domain name server... bind [OK]

# nslookup www.mynewsite.ie localhost

Server: localhost

Address: 127.0.0.1#53

Name: [www.mynewsite.ie](http://www.example.net)

Address: 192.168.0.7

If the BIND reload fails, it will say so on the command line. If that command-line output doesn’t tell you why BIND refuses to reload, you can view /var/log/syslog for clues.

## DNS Slave

A DNS slave is a DNS server that retrieves its zone information from a different DNS server known as its master. In fact, a BIND server can act as a master for one zone and a slave for a different zone. Having a master and one or more slave servers greatly simplifies your DNS administration because you have to update zone information only on the master and it automatically propagates to the slaves. When you update a zone on the master server and increment the serial number inside that zone, once BIND reloads, all of the slaves get notified that the zone has changed and will automatically pull down the updates.

Since a slave DNS server retrieves its zone information from the master, its configuration is much simpler. Provided the master is configured to allow zone transfers from the slave, on the slave you simply have to add an entry to **named.conf.default-zones**. For our example, let’s assume I wanted to add a second name server, ns2.mynewsite.ie at 192.168.0.10.

## Configure the Master Server

On my master server, 192.168.0.5, I would need to edit my **db.mynewsite.ie** file and add the references to ns2.example .net:

;

; BIND data file for mynewsite.ie

;

$TTL 1d

@ IN SOA ns1.mynewsite.ie. root.mynewsite.ie. (

3 ;Serial

604800 ;Refresh

86400 ;Retry

2419200 ;Expire

604800 ) ;Negative Cache TTL

;

@ IN NS ns1.mynewsite.ie.

@ IN NS ns2.mynewsite.ie.

ns1 IN A 192.168.0.5

ns2 IN A 192.168.0.10

www IN A 192.168.0.7

gw IN A 192.168.0.1

Now I need to add a line to the mynewsite.ie configuration in **named.conf.default-zones** so that it allows zone transfers from 192.168.0.10, so mynewsite.ie’s entry in **named.conf.default-zones** becomes

zone "mynewsite.ie" {type master;

file "/etc/bind/db.mynewsite.ie"; allow-transfer {192.168.0.10;};

};

## Configure the Slave Server

On the slave server I install the bind9 package either through **tasksel** or **apt-get,** and then all I need to do is add a slave entry for mynewsite.ie at  
the bottom of **/etc/bind/ named.conf.default-zones**:

zone "mynewsite.ie" { type slave;

file "/var/cache/bind/db.mynewsite.ie"; masters {192.168.0.5;};

};

Reload BIND with **service bind9 reload**. Once BIND reloads, it will immediately attempt a zone transfer for mynewsite.ie from 192.168.0.5. When it completes the zone transfer, if I check the /var/cache/bind/ directory I will see that the db.mynewsite.ie zone file was created there. Now the next time I want to make a change to db.mynewsite.ie, I just have to change the file on the master, update the serial number, and then reload BIND on the master. The slave will automatically get the updates.

## Manage BIND with rndc

So far I have reloaded BIND with the bind9 init script. Ubuntu includes another tool named rndc that helps you with BIND administration. For instance, to reload the BIND configuration using rndc, I would type

# rndc reload

You can pass a number of other arguments to rndc to get more information about the BIND service or submit commands to it. One useful aspect of the rndc command is that if you want an administrator to be able to update DNS but not have full root privileges, you can give that person sudo access to the rndc command. If you type rndc with no arguments, you will get a help page that lists the available commands. Here are some of the more interesting ones:

* **reload**—reloads all configuration files and zones. If you specify a specific zone after the reload command, it will reload only that zone.
* **retransfer zone**—retransfers the zone whether the serial number has been incremented or not.
* **reconfig**—like the reload command, except it only reloads named.conf and any new zones.
* **flush**—flushes all of a server’s caches. This is handy on a caching name server if it is holding on to a stale IP address that is no longer valid.
* **status**—outputs some statistics about the current status of the BIND process, including how many zones it is managing and some statistics on its current workload.

There are numerous resources available if you want more information about BIND configuration file syntax or DNS administration in general. The bind9-doc package includes a series of documentation files under /usr/share/doc/bind9-doc/ that are a great place to start. For instance, to view the first chapter of the BIND version 9 HTML manual, type w3m /usr/share/doc/bind9-doc/arm/Bv9ARM.ch01.html.